HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science
and Engineering

Jaakko Väyrynen

Learning linguistic features from natural text data by independent component analysis

Master's Thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Technology

Espoo, 13.1.2005

Supervisor: Prof. Timo Honkela
Instructor: Doc. Aapo Hyvärinen

TEKNILLINEN KORKEAKOULU
Tietotekniikan osasto

DIPLOMITYÖN TIIVISTELMÄ

Tekijä **Jaakko Väyrynen**	Päiväys **13.1.2005** Sivumäärä **63**
Työn nimi **Kielipiirteiden oppiminen luonnollisesta tekstistä riippumattomien komponenttien analyysilla**	
Professuuri **Informaatiotekniikka**	Koodi **T-61**
Pääaine **Informaatiotekniikka**	
Sivuaine **Kieliteknologia**	
Työn valvoja **Prof. Timo Honkela**	
Työn ohjaaja **Dos. Aapo Hyvärinen**	

Luonnollisen kielen analysointi on tärkeä tutkimusaihe kieliteknologian kannalta. Symbolinen kirjoittu kieli voidaan koodata numeerisessa muodossa ja analysoida käyttäen tilastollisia signaalinkäsittelymenetelmiä. Tässä diplomityössä oletetaan sanojen käytön, erityisesti sanojen esiintymistaajuuksien konteksteissa, sisältävän tilastollisilla menetelmillä irrotettavaa kielellistä informaatiota. Riippumattomien komponenttien analyysia, erästä ohjaamattoman oppimisen menetelmää sokeaan lähde-erotteluun, sovelletaan piirreirrotukseen tekstikorpuksesta. Vertailu löydettyjen piirteiden ja perinteisten syntaktisten sanakategorioiden samankaltaisuuden välillä osoitti, että riippumattomien komponenttien analyysi irrotti piirteitä jotka muistuttavat enemmän kielellisiä kategorioita kuin pääkomponenttianalyysillä irrotetut piirteet.

Avainsanat
riippumattomien komponenttien analyysi, luonnollisen kielen tilastollinen käsittely

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF MASTER'S THESIS
Department of Computer Science and Engineering

Author	Date
Jaakko Väyrynen	**13.1.2005**
	Pages **63**
Title of the thesis **Learning linguistic features from natural text data by independent component analysis**	
Professorship **Computer and Information Science**	Professorship Code **T-61**
Major **Computer and Information Science**	
Minor **Language Technology**	
Supervisor **Prof. Timo Honkela**	
Instructor **Doc. Aapo Hyvärinen**	

The analysis of natural language is an important field for language technology. The symbolic nature of written language can be encoded in numeric form and analyzed using statistical signal processing methods. In this thesis, it is assumed that word usage statistics, namely word frequencies in different contexts, contain linguistic information that can be extracted using statistical feature extraction methods. Independent component analysis, an unsupervised statistical method for blind source separation, is applied to extracting features for words using a text corpus. A study between the closeness of match between the emergent features and traditional syntactic word categories for words shows that independent component analysis extracts features that resemble more linguistic categories than features extracted with principal component analysis.

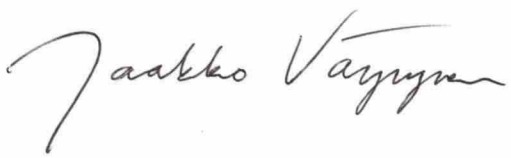
Keywords
independent component analysis, statistical natural language processing

Preface

This thesis has been written in the Laboratory of Computer and Information Science at Helsinki University of Technology. The facilities and the inspiring environment has been provided by the Neural Network Research Centre, one of the Centres of Excellences elected by the Academy of Finland.

I would like to express my gratitude to my supervisor Professor Timo Honkela and my instructor Docent Aapo Hyvärinen for the huge amount of encouragement and advice they have provided. I would also like to mention Doctor Jarmo Hurri and the members of the Adaptive Cognitive Systems group.

Espoo, 13.1.2005



Jaakko Väyrynen

Contents

Abbreviations

BSS	Blind source separation
CFG	Context free grammar
(H)MM	(Hidden) Markov model
ICA	Independent component analysis
LSA	Latent semantic analysis
ML	Maximum likelihood
NLP	Natural language processing
NMF	Non-negative matrix factorization
PCA	Principal component analysis
PCFG	Probabilistic context free grammar
RP	Random projection
SOM	Self-organizing map
SVD	Singular value decomposition

Chapter 1

Introduction

Statistical methods in natural language processing have been an active research field in the last ten years [20, 7, 32, 43, 38, 10, 15]. Increased computational power and storage capabilities have made storing and analyzing large electronic language corpora feasible.

Traditional linguistic methods prefer rule-based and hand-crafted models, that are subjective of their creators knowledge and intentions. Statistics provides a more objective method to model languages flexibly and adaptively.[1] Unsupervised methods, such as independent component analysis (ICA) [24], that is studied in detail in this thesis, needs only examples of language usage for the learning process. The estimated models will reflect the language, vocabulary and style of the data used. Simple statistical models may not capture all the subtleties of the language, but are easier to create and adjust to specific needs than manually created models.

Language can be modeled on several linguistic levels, for instance, on phonetic, orthographic, morphologic, syntactic, semantic, and pragmatic levels, ranging from acoustics to intentions. The models can increase our knowledge of language based communication and the process of learning languages. This will help creating applications, such as automatic translation, speech processing, information retrieval and other natural language processing tasks.

Some of the statistical methods previously applied to the problem of analyzing and modeling natural language include the self-organizing map (SOM) [27], singular value decomposition (SVD) [18] and several clustering methods [7, 20,

[1] The remaining subjective aspects include the choice of corpora, processing methods and parameters.

13, 32, 2, 33]. The methods have been used, for instance, to create automatic categorization and vector models for words.

Independent component analysis is an unsupervised method for extracting statistically independent components from observed mixtures of signals. The assumption is that the observations are linear combinations of source signals in a noise free environment. In contrast to singular value decomposition, which uses only second order statistics, independent component analysis uses information from higher order statistics [24].

1.1 Problem setting

Data modeling using statistical methods is an active research area. Increased calculation power and electronically stored large data sets has made statistical analysis feasible for multimedia signals such as sound, text and video. The analysis of natural language is a complex task, and several methods have already been applied to it. In this thesis, independent component analysis is applied to the analysis of written natural language. The emerging features are considered to represent linguistic information in the corpora.

Traditional linguistic features depend on the language and are manually determined. Several sets of word categories exist for different corpora and languages. Applying methods to several languages and corpora with manually constructed components is not straightforward. Automatic extraction of linguistic features should benefit the analysis of natural language and natural language processing in general.

1.2 Aim of the Thesis

The aim of this thesis is to further study the emerging linguistic representation found by independent component analysis, an unsupervised statistical method that has only information on word frequencies in different contexts. The learned features provide a distributed representation of words, where a word can have several features simultaneously. This thesis studies how well the learned features capture linguistic information by the closeness of match between the learned features and traditional manually determined linguistic categories.

1.3 Contributions of the Thesis

The work in this thesis is based on the paper by Honkela et al. [19] where the idea of applying independent component analysis to extract linguistic information from contextual information was first introduced. In this thesis, the framework and parameter selections have been studied in more detail by the author, and a method for comparing the extracted features and traditional linguistic categories has been studied.

1.4 Structure of the Thesis

The structure of this thesis is the following.

After introducing the problem in Chapter 1, the basic background information if covered in Chapter 2. Natural language processing and information theory are reviewed from the point of statistical natural language processing. The applied computational methods, principal component analysis, singular value decomposition and independent component analysis are introduced.

Related topics to this thesis are discussed in Chapter 3. Word categorization and contextual information are themes closely related to this work. Self-organizing map, latent semantic analysis (LSA) [32] and other statistical models are discussed as alternatives to independent component analysis.

In Chapter 4, the integral part of this thesis, applying independent component analysis to word contexts, is introduced. The method, the meaning of different parameter and the selection of learning data are discussed.

A survey of the data sets is done in Chapter 5 and the conducted experiments and results are explained in Chapter 6 in detail.

Finally, in Chapter 7, conclusions about the work are drawn, and the future and meaning of the work are discussed.

Chapter 2

Background

Statistical natural language processing can be approached with many tools, depending on the view of what language is and how it should be represented. Information theory provides the means to analyse language as a stochastic communication system. With a stochastic approach, uncorrelatedness and independence are important concepts. Statistical methods, such as principal component analysis, singular value decomposition and independent component analysis, play a major part in statistical natural language processing and in this work.

2.1 Natural language processing

The optimistic visions of human-quality automatic translation, speech recognition and synthesis from the early days of natural language processing were soon discarded, when several of the problems revealed to be difficult to solve.

A late trend in natural language processing is to use statistical and adaptive methods that are taught on real examples of language usage. Digital storage and computational power are inexpensive enough to make it feasible to use large corpora that are needed for a statistical representation of languages. The use of statistical methods can be motivated by how the human brain learns language without explicit knowledge of grammar or other linguistic information.

Models constructed with statistical methods and tools, such as information theory, may give insight to cognitive processes. Statistical and unsupervised models may also result in systems that, for instance, can adapt and handle

errors better than hand-crafted and non-stochastic systems. The diversity of language requires models for different languages and dialects as well as for spoken and written representations. Languages are also studied and modeled on several levels varying from acoustics to intentions.

2.1.1 Linguistics

Language is an essential part of the human world. Linguistic science tries to explain how languages are acquired, produced and understood. A major part of the grammatically well-formed phrases can be generated by using simple syntactic rules that combine linguistic structures, but they fail to explain everything. As the linguist Sapir stated, "all grammars leak" [42].

Linguists have approached the problem of modeling language by trying to understand the language in human mind (rationalist approach) and the actual use of the language (empiricist approach) [38].

2.1.2 Categorization

Categories as abstractions are need for the interpretation of the empirical word. Aristotle (384–332 B.C.) already distinguished ten categories. The most common are 1. Items (objects). 2. Qualities (properties). 3. States or state changes. 4. Relations (spatial, temporal, and other). The three first categories correspond to linguistic categories of nouns, adjectives and verbs. The fourth category can be represented with syntax, prepositions, postpositions, endings and inflections [40].

The membership in a category can be chosen in many different ways. Each item could belong to only one or to several categories simultaneously. Likewise, belonging to a category can be crisp or graded. In a crisp membership, an object either belongs to category, or it doesn't belong to it. In a graded membership, being in a category is indicated, for instance, by a real number, that tells how much the objects resembles the category.

Categorization is also a major phenomenon in language in general. It can be seen in the way languages assign different names to different things. There are for instance animals, emotions and colors. There are also hierarchies of categories and sub-categories, for instance, different names for colors and different species of animals. Categorization has been modeled for instance by shared features of members and by prototype theory [30].

One linguistic task is categorization, in which things are labeled as belonging to different classes. Closely related NLP tasks include author and language identification, text categorization by topic, and disambiguation. In all the tasks a class is assigned for an object.

Categorization is also related to perception. A phenomenon called categorical perception tells how the differences between categories and inside a category are perceived, for instance, how spoken utterances of vowel categories are perceived [17]. The differences between utterances from different categories are perceived more clearly, whereas the differences inside a category are not so easily distinguished. Similar phenomenon has been observed in the color perception of the visual system [39].

2.2 Information Theory

Language is a form of communication, in which messages are expressed in a medium, be it speech, sign language or writing, and transferred to others, who try to understand the conveyed ideas by what they understand themselves. To express a message, first it must be represented in a language. The language might have a set of symbols, and rules and meanings for combining the symbols. The symbols are furthermore relayed to the receiver, for instance, in speech the message is transformed into sound waves by the acoustic properties of the vocal cords and the vocal tract. The receiver must interpret the heard sounds as words in order to receive the symbols. To understand the meaning of the message, the receiver must interpret the received symbols in a meaningful context. On the other hand, it has been pointed out that information theory does not provide the means to handle semantics.

Information theory has been the basis for understanding communication systems. In the following, the essentials of information theory is introduced. Information theory has also contributed to independent component analysis by giving measures of independence.

2.2.1 Entropy and information

Entropy $H(X)$ measures the average information of the finite set of possible outcomes x_i of the random variable X. The uncertainty of the event x_i is measured as the probability $p(x_i) = P(X = x_i)$. The entropy $H(X)$ of such

discrete random variable X is defined by the formula

$$H(X) = -\sum_{x \in X} p(x) \log p(x) = \mathrm{E}\left\{\frac{1}{\log p(x)}\right\} \tag{2.1}$$

where $p \log p$ goes to zero in the limit $p \to 0$. If the base of the logarithm is two, the quantity is expressed in *bits*. In a discrete system, entropy is maximized when all the probabilities $p(x)$ are equal. Entropy is minimized when a single outcome is fixed, i.e., it has probability one, and all the other outcomes have zero probability.

For continuous variables there is an infinite number of outcomes, and the formula

$$H(X) = -\int p(x) \log p(x) \mathrm{d}x = \mathrm{E}\left\{\frac{1}{\log p(x)}\right\} \tag{2.2}$$

for differential entropy is used. An important result is that a Gaussian variable has the largest entropy among all continuous random variables with fixed variance.

2.2.2 Joint entropy and conditional entropy

The amount of information in a pair of random variables X, Y is measured with the joint entropy

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(x,y) = H(X) + H(Y|X) \tag{2.3}$$

The conditional entropy

$$H(Y|X) = -\sum_{x \in X} \sum_{y \in Y} p(x|y) \log p(x|y) = \sum_{x \in X} p(x) H(Y|X = x) \tag{2.4}$$

expresses the unknown information about Y when X is known.

2.2.3 Mutual information

Mutual information

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{2.5}$$

measures the common information in the two random variables X and Y, i.e., the amount of information one random variable contains about the another. It is a symmetric and non-negative measure, that is zero when the two variables are independent. Independence is discussed in Section 2.3.2. The relationship between mutual information I and entropy H is illustrated in Figure 2.1. Mutual information has been applied to automatic word categorization [33] and independent component analysis [24].

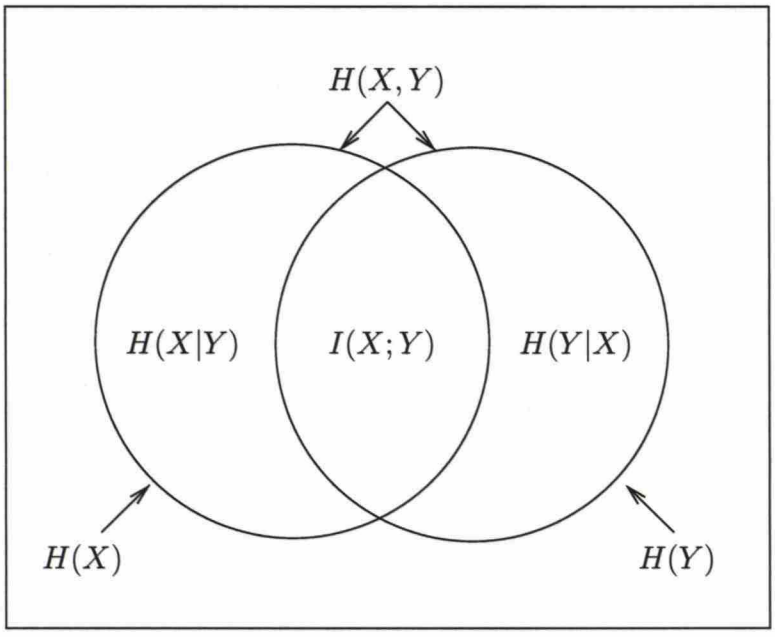


Figure 2.1: Mutual information $I(X;Y)$ measures the common information between two random variables, X and Y. The information of a random variable is measured with entropy H.

2.2.4 Negentropy

Given the information that a Gaussian variable maximizes differential entropy, it can be seen that entropy can be used as measure of non-Gaussianity. A more approachable measure is negentropy

$$J(X) = H(X_{Gauss}) - H(X) \tag{2.6}$$

where X_{Gauss} is a Gaussian random vector of the same covariance matrix as X. Negentropy $J(X)$ is always nonnegative and zero only when X is Gaussian. Unfortunately, negentropy is computationally too demanding to be used directly, so approximations are commonly used.

2.2.5 Natural language processing

Shannon's models of communication over noisy channels are derived using information theory. Language can be seen as communication through a noisy channel, and several statistical NLP tasks, such as speech recognition, can be seen as decoding problems, where the original signal is being decoded from the noisy output with the help of a model for the encoding process. In speech recognition, the problem is to decode the original word sequence from the speech signal using an acoustic model that represents the encoding of the word sequence to the speech signal. Similarly, ideas can be conveying using a word sequence as the medium. The several layers of linguistic analysis, mentioned in Chapter 1, makes natural language processing a demanding task.

2.3 Uncorrelatedness and independence

In the following, two measures, uncorrelatedness and statistical independence, are discussed as measures for comparing random variables statistically. Uncorrelatedness is based on second order central-moment statistics, and is a special case of statistical independence, that considers higher-order statistics.

2.3.1 Uncorrelatedness

For two random vectors $\mathbf{x}$ and $\mathbf{y}$ to be uncorrelated, their cross-covariance matrix $\mathbf{C}_{\mathbf{xy}}$ must be a zero matrix:

$$\mathbf{C}_{\mathbf{xy}} = \mathrm{E}\left\{(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{y} - \mathbf{m}_{\mathbf{y}})^T\right\} = \mathbf{0}, \tag{2.7}$$

where $\mathbf{m}_{\mathbf{x}} = \mathrm{E}\{\mathbf{x}\}$ and $\mathbf{m}_{\mathbf{y}} = \mathrm{E}\{\mathbf{y}\}$ are the mean vectors of $\mathbf{x}$ and $\mathbf{y}$, respectively. For zero-mean variables, zero covariance is equivalent to zero correlation. The requirement of Equation 2.7 is equivalent to the condition

$$\mathbf{R}_{\mathbf{xy}} = \mathrm{E}\left\{\mathbf{x}\mathbf{y}^T\right\} = \mathrm{E}\{\mathbf{x}\}\,\mathrm{E}\left\{\mathbf{y}^T\right\} = \mathbf{m}_{\mathbf{x}}\mathbf{m}_{\mathbf{y}}^T \tag{2.8}$$

on the correlation matrix $R_{\mathbf{xy}}$.

Considering the correlations between the components of a random vector $\mathbf{x}$ defined by the covariance matrix $\mathbf{C}_{\mathbf{x}}$

$$\mathbf{C}_{\mathbf{x}} = \mathrm{E}\left\{(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T\right\} \tag{2.9}$$

the requirement of Equation 2.7 can never be met. At best, the correlation matrix is a diagonal matrix

$$\mathbf{C_x} = [\hat{c}_{ij}] = \text{diag}\left(\sigma_{x_1}^2, \sigma_{x_2}^2, \ldots\right) \tag{2.10}$$

with the variances $\sigma_{x_i}^2$ of the components x_i in the diagonal values c_{ii}. The components c_{ij} of the covariance matrix $\mathbf{C_x}$ are the covariances between components x_i and x_j, which is zero if the components are uncorrelated. For real valued signals the covariances are real, and the symmetry of the covariances matrix follows from the symmetry

$$\text{cov}\,(x, y) = \text{cov}\,(y, x) \tag{2.11}$$

property of covariance for real signals.

Random vectors having zero-mean and an identity matrix as the covariance matrix are called *white* or *sphered*. Thus the requirements for whiteness are

$$\mathbf{m_x} = \mathbf{0}, \qquad \mathbf{R_x} = \mathbf{C_x} = \mathbf{I} \tag{2.12}$$

where $\mathbf{I}$ is an identity matrix.

2.3.2 Statistical independence

Independent component analysis relies on the concept of statistical independence. For the random variables x and y to be independent, knowing the value of x must not give any information about the value of y, and vice versa. Examples of independent processes are the value of a dice thrown and of a coin tossed, or speech signal and background noise originating from a ventilation system, in which the value of the dice throw does not affect the outcome of the coin toss.

A necessary and sufficient condition for the independence of two scalar random variables is that their joint probability density $p_{x,y}(x, y)$ is factorizable to the marginal densities:

$$p_{x,y}(x, y) = p_x(x)p_y(y), \tag{2.13}$$

and the definition of Equation 2.13 generalizes to random vectors and multiple variables:

$$p_{\mathbf{x},\mathbf{y},\mathbf{z},\ldots}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots) = p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}}(\mathbf{y})p_{\mathbf{z}}(\mathbf{z}) \ldots . \tag{2.14}$$

Independent variables satisfy

$$\text{E}\left\{\mathbf{g_x}(\mathbf{x})\mathbf{g_y}(\mathbf{y})\mathbf{g_z}(\mathbf{z}) \ldots\right\} = \text{E}\left\{\mathbf{g_x}(\mathbf{x})\right\} \text{E}\left\{\mathbf{g_y}(\mathbf{y})\right\} \text{E}\left\{\mathbf{g_z}(\mathbf{z})\right\} \ldots \tag{2.15}$$

for any absolutely integrable functions $\mathbf{g}(\cdot)$. Comparing Equations 2.15 and 2.8, it can be seen that uncorrelatedness follows from independence with linear functions $\mathbf{g}(\cdot)$, and independence is a much stronger property than uncorrelatedness.

2.4 Principal Component Analysis

Principal component analysis [18] is a statistical method that finds an orthogonal basis which transforms possibly correlated variables into a number of uncorrelated variables without losing any information. The directions of the basis vectors, called principal components, give the directions of the largest variances in the data. The data vectors are projected onto the orthogonal basis, where the projected vectors are uncorrelated. Uncorrelatedness is defined in Section 2.3.1.

Additionally, principal component analysis can be used to reduce the dimension of the data optimally in the mean square sense, by using only the principal components with the largest variances, i.e., modeling the data with only a subset of the new basis vectors. Projecting the data vectors to the new basis gives projected vectors that are uncorrelated and possibly lower-dimensional than the original vectors. The projected vectors can be analyzed as they are, or they can be used as input to further processing. For instance, independent component analysis, reviewed in Section 2.6.4 can be seen as an extension of principal component analysis.

The process of finding the principal components can be thought as first taking the direction of the largest variance in the data and naming it the first principal component. Further principal components are calculated on a subspace that is orthogonal to the principal components already found. Figure 2.2 shows the principal components of a two-dimensional Gaussian variable.

A closed form solution to PCA can be derived using the eigenvalues λ_k of the covariance matrix $\mathbf{C}_\mathbf{x}$ of a random vector population $\mathbf{x} = (x_1, \ldots, x_m)^T$. The eigenvectors are in fact the directions of the principal components. The covariance matrix of Equation 2.7 and the mean vector $\mathbf{m}_\mathbf{x}$ can be estimated

from sample vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ as

$$\hat{\mathbf{m}}_\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{2.16}$$

$$\hat{\mathbf{C}}_\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mathbf{m}}_\mathbf{x})(\mathbf{x}_i - \hat{\mathbf{m}}_\mathbf{x})^T \tag{2.17}$$

and the estimates $\hat{\mathbf{m}}_\mathbf{x}$ and $\hat{\mathbf{C}}_\mathbf{x}$ can be used in the analytical formulas.

Following from the properties of the covariance matrix $\mathbf{C}_\mathbf{x}$, the eigenvalue decomposition

$$\mathbf{C}_\mathbf{x}\mathbf{Q} = \mathbf{\Lambda}\mathbf{Q} \tag{2.18}$$

can then always be calculated, where $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_m)$ is a $m \times m$ matrix with the eigenvectors $\mathbf{q}_i$ as columns, and $\mathbf{\Lambda} = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_m)$ is a diagonal matrix with the corresponding eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m$.

With $\mathbf{Q}$ being orthogonal, its inverse is its transpose, and Equation 2.18 can be rewritten as the orthogonal similarity transformation

$$\mathbf{Q}^T\mathbf{C}_\mathbf{x}\mathbf{Q} = \mathbf{\Lambda} \tag{2.19}$$

by multiplying with $\mathbf{Q}^{-1} = \mathbf{Q}^T$ on the left.

Now the linear transformation

$$\begin{aligned} \mathbf{V} &= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T \\ \mathbf{z} &= \mathbf{V}(\mathbf{x} - \mathbf{m}_\mathbf{x}) \end{aligned} \tag{2.20}$$

whitens the data so that the components are uncorrelated and variances equal unity, i.e., the covariance matrix is an identity matrix $\mathbf{I}$. This can be verified by writing the covariance matrix $\mathbf{C}_\mathbf{z}$ for the transformed vectors $\mathbf{z}$:

$$\begin{aligned} \mathbf{C}_\mathbf{z} &= \mathrm{E}\left\{(\mathbf{z} - \mathrm{E}\{\mathbf{z}\})(\mathbf{z} - \mathrm{E}\{\mathbf{z}\})^T\right\} = \mathrm{E}\left\{\mathbf{z}\mathbf{z}^T\right\} \\ &= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\mathrm{E}\left\{\mathbf{x}\mathbf{x}^T\right\}\mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}} \\ &= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\mathbf{C}_\mathbf{x}\mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{\Lambda}\mathbf{\Lambda}^{-\frac{1}{2}} \\ &= \mathbf{I} \end{aligned} \tag{2.21}$$

and it can be seen that the transformed vectors $\mathbf{z}$ satisfy the whiteness conditions in Equation 2.12 of identity covariance matrix and zero mean.

The reverse transformation to the original space is

$$\begin{aligned} \mathbf{V}^{-1} &= \mathbf{Q}\mathbf{\Lambda}^{\frac{1}{2}} \\ \mathbf{x} &= \mathbf{V}^{-1}\mathbf{z} + \mathbf{m}_\mathbf{x} \end{aligned} \tag{2.22}$$

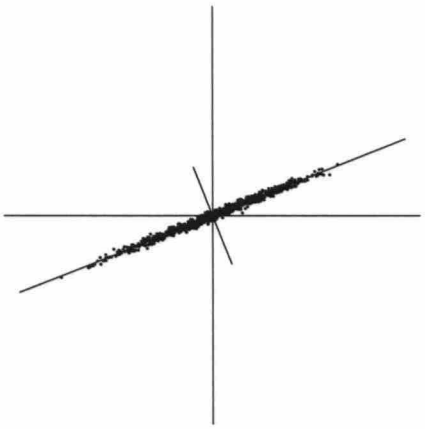


Figure 2.2: A two-dimensional dataset and the estimated principal components. Projecting the data to the principal component with the largest variance (longer line) preserves most of the structure in the data.

Dimensionality reduction can be accomplished by selecting only the first n ($n < m$) largest eigenvalues $\hat{\mathbf{\Lambda}} = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$ and the corresponding eigenvectors $\hat{\mathbf{Q}} = (\mathbf{q}_1, \ldots, \mathbf{q}_n)$. This is equivalent to selecting the principal components responsible for the largest variances in the data. This is illustrated in Figure 2.2, where only one principal component models the two-dimensional data quite well. The approximation error, ε, can be shown to be equal to the sum of the eigenvalues of the principal components left out

$$\varepsilon = \sum_{j=n+1}^{m} \lambda_j \tag{2.23}$$

and it minimizes the mean square error between the original vector $\mathbf{x}$ and it's approximation $\hat{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{z} + \mathbf{m}_\mathbf{x}$ in the reduced orthogonal base with the given number of principal components.

2.5 Singular value decomposition

In singular value decomposition [18], the latent structure of vectors is used to decompose a matrix $\mathbf{X}$ of size $t \times d$ into the product of three other matrices

$$\mathbf{X} = \mathbf{USV}^T \tag{2.24}$$

as illustrated in Fig. 2.3. The matrices $\mathbf{U}$ and $\mathbf{V}$ contain the left and right singular vectors, respectively, and $\mathbf{S}$ is a diagonal $m \times m$ matrix of singular

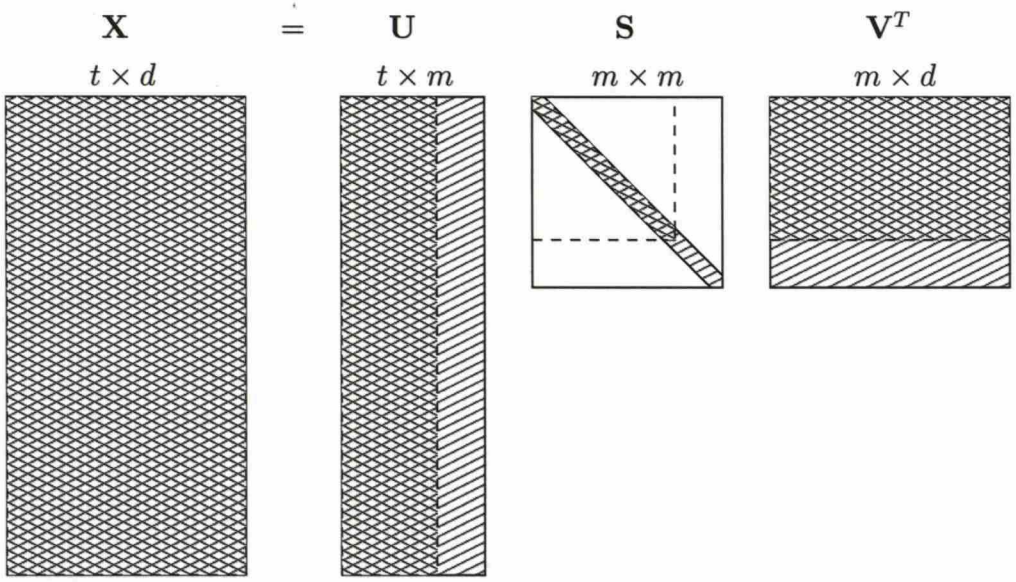


Figure 2.3: Singular value decomposition decomposes the matrix $\mathbf{X}$ into three matrices $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$ using the latent structure in the vectors of $\mathbf{X}$. The diagonal matrix $\mathbf{S}$ has the singular values in descending order. Optimal reconstruction in the least-squares sense is accomplished by dropping out the smallest singular values and the corresponding vectors from the left and right singular matrices. This is illustrated in the shading of the boxes representing the matrices.

values in descending order. The matrices $\mathbf{U}$ and $\mathbf{V}$ have orthogonal, unit-length columns and describe the original row and column entities with the orthogonal factors. The singular values scale the components appropriately, so that the original matrix is reconstructed [32]. Singular value decomposition can always be calculated and is unique up to some permutations in the product matrices [13] and the number of factors m is at most $\min(t, d)$.

The dimensionality of the factorization can be reduced optimally in the least-squares sense by dropping out small singular values in the reconstruction. In Figure 2.3, dimension reduction is illustrated with lighter shading of the dropped out regions.

2.5.1 Relation to principal component analysis

In principal component analysis, explained in Section 2.4, the square symmetric covariance matrix $\mathbf{C}_\mathbf{X} = \mathbf{X}\mathbf{X}^T$, is decomposed into the product $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$.

The eigenvectors of the covariance matrix are the columns of $\mathbf{Q}$, and the eigenvalues are the diagonal values in $\mathbf{\Lambda}$.

Singular value decomposition $\mathbf{X} = \mathbf{USV}^T$ is directly related to the eigenanalysis done by principal component analysis. The left singular vectors $\mathbf{U}$ are the eigenvectors of the square symmetric matrix $\mathbf{XX}^T$ and the right singular vectors $\mathbf{V}$ are the eigenvectors of the matrix $\mathbf{X}^T\mathbf{X}$. In both cases the eigenvalues $\mathbf{\Lambda}$ are proportional to the squared singular values $\mathbf{S}^2$.

2.6 Independent Component Analysis

Estimating original signals from observations of mixture signals without any information about the original signals or the mixing process is called blind source separation (BSS). In the following, independent component analysis [12, 24], an unsupervised statistical method for blind signal separation is introduced. The basic principles behind independent component analysis, and how the independence of signals is measured are explained. A short overview of an estimation algorithm is given.

2.6.1 Introduction

Independent component analysis does blind source separation by trying to estimate a linear mixing matrix and the original signals using observations of the mixture signals under the assumption that the sources are statistically independent. ICA has been applied successfully to biomedical signal processing, telecommunications, speech processing, image feature extraction, and classification systems [3, 48, 45, 45, 1, 16, 6, 34, 4].

An often mentioned application with independent component analysis is the so-called "cocktail-party problem", in which the objective is to separate individual sound sources in a room. The observations are recorded in different locations of the room with each microphone recording only a cacophony of voices. A simplified view of the mixing process creates each observation as a linear weighted sum of the original sources. The weights are determined solely by the attenuation of the sound, i.e., distance between the source and the microphone. Assuming statistical independence of the sources, independent component analysis can be used to estimate the original sound sources.

The "cocktail-party problem" is a good example of how independent compo-

nent analysis is usually applied by assuming simple mixing process and statistical independence of the sources, when neither may not hold. In the problem, the mixing process is simplified by neglecting important acoustic effects, such as time delay and reverberation, as well as the possibility of people moving. Also, statistical independence may not hold in this case because, for instance, people may take turns talking and listening.

An example of signal separation with ICA is given in Figure 2.4, where three original signals are mixed with random weights to create three mixture signals. The extracted independent components are quite close to the original signals, except for the signs and the order of signals. It should be noted that even though the original signals are periodic, the ICA algorithm does not take advantage of the time information. Actually, the order of the samples should have no effect on the results.

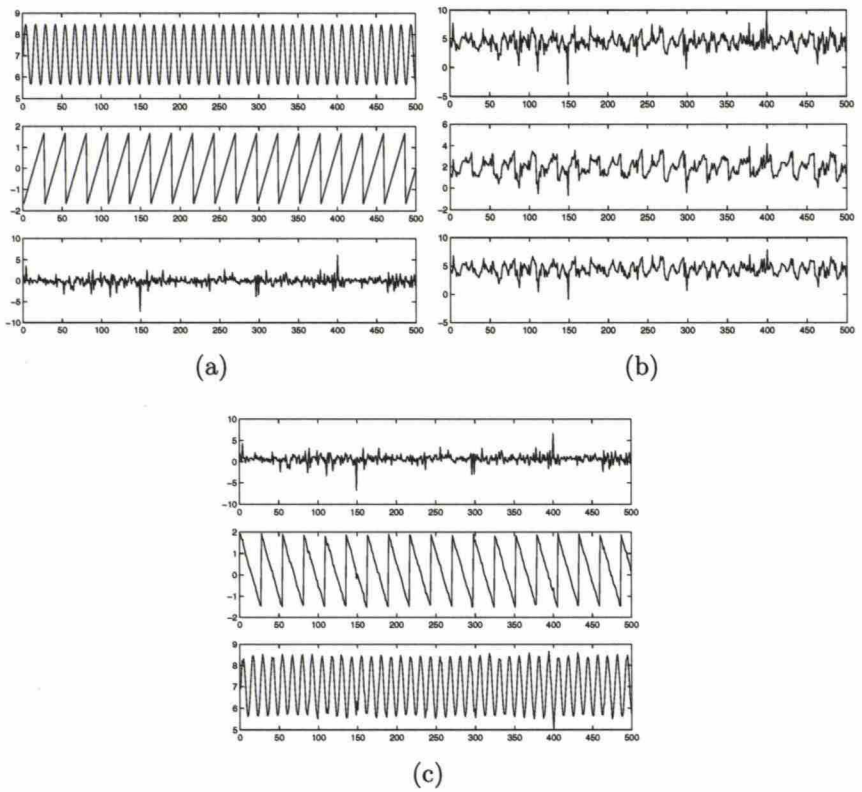


Figure 2.4: a) Original sine, sawtooth and impulse noise signals. b) Mixture signals created by combining the original signals with random weights. c) Independent components extracted from the mixture signals. The original signals are found quite well by ICA, except for the order and the signs of the sources.

2.6.2 ICA framework

In matrix form, the ICA framework is usually defined as the linear noise-free generative model

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{2.25}$$

where the latent source signals or independent components, are represented by the random variable vector $\mathbf{s} = (s_1, \ldots, s_n)^T$. The observed mixture signals $\mathbf{x} = (x_1, \ldots, x_n)^T$ are generated by multiplying the sources with a constant $n \times n$ square mixing-matrix $\mathbf{A}$. The source signals s_i are assumed to have zero mean, which means that the observed signal components x_j are zero mean also.

The generative model of Equation 2.25 can also be written as

$$\mathbf{x} = \sum_{k=1}^{n} \mathbf{a}_k s_k, \tag{2.26}$$

where the columns $\mathbf{a}_k$ of the mixing matrix $\mathbf{A}$, give the basis where the observations are represented.

Given samples of observations $\mathbf{x}_i$, $1 \leq i \leq N$, and the information that the sources s_j are statistically independent, the problem is to estimate both the mixing-matrix $\mathbf{A}$ and the sources $\mathbf{s}$ simultaneously.

If the estimated mixing matrix $\mathbf{A}$ is known, the sources can be estimated by multiplying the observed signals with the inverse of the estimated mixing matrix $W = \mathbf{A}^{-1}$:

$$\mathbf{s} = \mathbf{y} = \mathbf{A}^{-1}\mathbf{y} = \mathbf{W}\mathbf{x}. \tag{2.27}$$

Assumptions for ICA

The sources $\mathbf{s}_k$ can be assumed to be zero mean

$$\mathrm{E}\{\mathbf{s}_k\} = \mathbf{0}, \tag{2.28}$$

without any loss of generality. This makes the mixed observations $\mathbf{x}$ zero mean also and the assumption reduces the complexity of the estimation algorithms.

The sources are assumed to be statistically independent. Section 2.3.2 introduces the requirements for statistical independence. The assumption seems

quite hard to meet in most processes. Instead, independent component analysis in interpreted as finding the most independent components. In an another view, independent component analysis is seen as the maximization of the sparseness of the sources [24], i.e., the sources should have only a few entries that differ significantly from zero.

One more assumption for ICA is that only one of the sources can be normally distributed [24]. Otherwise the distributions of the sources can be unknown. To see why ICA is impossible when there are more than one normal source, consider the case of two independent, white and normal variables y_1 and y_2. This is illustrated in Figure 2.5. The joint distribution

$$p(y_1, y_2) = \frac{1}{2\pi} \exp\left(-\frac{y_1^2 + y_2^2}{2}\right) \tag{2.29}$$

is also normal and completely symmetric, so the directions of the columns of the mixing matrix cannot be determined.

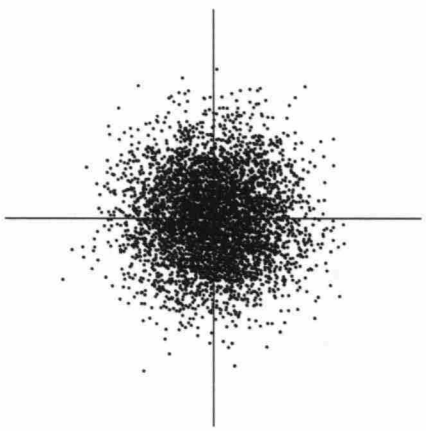


Figure 2.5: A distribution of two independent, white and normally distributed components.

Ambiguities of ICA

There are certain ambiguities, namely the order, signs and magnitudes of the sources, that cannot be inferred with independent component analysis. These follow from the linearity of Equation 2.25, and the fact that the sources and the mixing matrix are estimated simultaneously.

The order of the summation in Equation 2.26 can be modified without changing the result, i.e., the order of the sources cannot be determined within the

ICA framework. This differs from principal component analysis, where the eigenvalues define an ordering of the eigenvectors. The ambiguity of the order of the sources can be written as a multiplication of the sources $\mathbf{s}$ with a permutation matrix $\mathbf{P}$ and the mixing matrix $\mathbf{A}$ with the inverse of the permutation matrix. Similarly, the sources can be scaled and the signs changed with a full-rank diagonal scaling matrix $\mathbf{M} = \operatorname{diag}(m_1, \ldots, m_n)$, as long as the mixing matrix is scaled appropriately. These ambiguities can be combined into a matrix $\mathbf{B} = \mathbf{PM}$ that scales, changes signs, and permutes the sources:

$$\mathbf{x} = (\mathbf{A}(\mathbf{PM})^{-1})(\mathbf{PMs}) = (\mathbf{AB}^{-1})\mathbf{Bs}. \tag{2.30}$$

From these properties it follows that the signs, variances and the relative importances of the sources cannot be determined. As the variances of the sources cannot be determined, they are usually scaled to unity, $\mathrm{E}\{s_k^2\} = 1$. These ambiguities are visible in the example Figure 2.4, where the estimated independent components do not have the same order nor the signs as the original signals.

Measuring independence

The fundamental task in ICA is to measure the independence of the estimated components $\mathbf{y} = \mathbf{Wx}$. There are several ways to do it. An intuitive approach is to search for maximally non-Gaussian components. This can be motivated by the central limit theorem, that says that the sum of a large number of independent random variables converges to a normal distribution. Then the less the components resemble normal distributions, the less constituent variables it has.

Non-Gaussianity can be measured, for instance, by kurtosis or by approximations of negentropy. Other methods include maximum likelihood estimation and minimizing the mutual information of the components, many of which have been shown to be equivalent [24].

One independent component can be estimated as $y = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w}$ needs to be determined. If $\mathbf{w}$ really is one of the independent components, it is one of the rows of the inverse of the mixing matrix $\mathbf{A}$.

Information theory and ICA

Information theory provides measures of independence, a fundamental function in independent component analysis. The independent components are

estimated by maximizing a function $F(s_1, \ldots, s_n)$ that measures independence of random variables, where the components s_j are calculated using the Equation 2.27. In Section 2.2, mutual information and negentropy were introduced as measures of independence.

2.6.3 Preprocessing

Preprocessing techniques are often necessary to modify the data to be suitable for the algorithm used. Computationally lighter methods, such as PCA, can also reduce the complexity of the problem by solving part of it, or reducing the dimension of the data [24].

Here the basic preprocessing methods are discussed. Centering and whitening make the estimation simpler by reducing the space of the problem. Dimensionality of the data can be reduced by principal component analysis to remove noise and achieve a square mixing matrix.

Centering

As mentioned earlier, the observations $\mathbf{x}$ can be assumed to be zero mean. Let's call the non-centered observation vectors $\mathbf{x}'$. Now the centered observations $\mathbf{x}$ are simply

$$\mathbf{x} = \mathbf{x}' - \mathrm{E}\{\mathbf{x}'\}, \tag{2.31}$$

and after the estimation of the zero mean sources $\mathbf{s}$, the mean can be restored

$$\mathbf{s}' = \mathbf{s} + \mathbf{W}\mathrm{E}\{\mathbf{x}'\} \tag{2.32}$$

to the sources.

Whitening

One method for simplifying the problem is called whitening, introduced in Section 2.3.1. In whitening, the variables are forced to be uncorrelated and to have unit variance. When the observations $\mathbf{x}$ are centered, uncorrelatedness equals to zero correlation. A whitening transformation $\mathbf{V}$ is always possible to achieve. One method to calculate the whitening transformation $\mathbf{V}$ is the eigenvalue decomposition, which is explained in more detail in Section 2.4. Writing Equation 2.25 for the whitened vectors

$$\mathbf{z} = \mathbf{V}\mathbf{x} = (\mathbf{V}\mathbf{A})\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s} \tag{2.33}$$

makes the whitened mixing matrix $\tilde{\mathbf{A}} = \mathbf{VA}$ orthogonal, as can seen from

$$\mathrm{E}\left\{\mathbf{z}\mathbf{z}^T\right\} = \tilde{\mathbf{A}}\mathrm{E}\left\{\mathbf{s}\mathbf{s}^T\right\}\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I} \tag{2.34}$$

This restricts the space of the mixing matrices to orthogonal matrices, and thus reduces the number of degrees of freedom in the problem. The orthogonality of the whitened mixing matrix $\tilde{\mathbf{A}}$ states that it is in fact a rotation matrix and whitening has estimated the independent components up to a rotation. Independent component analysis can thus be defined as principal component analysis followed by whitening and an orthogonal rotation. After PCA, whitening requires only scaling variances to one. This is illustrated in Figure 2.6, where ICA is illustrated as centering, whitening and rotation.

Dimension reduction

In addition to whitening, the dimensionality of the observations can be reduced with PCA, as explained in Section 2.4. Instead of using all the principal components when projecting the vectors, only the ones with the largest variances are used. Dimension reduction can be seen as a method for removing noise and making the ICA algorithm simpler by making the mixing matrix square.

2.6.4 FastICA

The FastICA [14] algorithm is a very efficient batch algorithm for independent component analysis implemented in MATLAB programming language. It can be seen as a fixed-point iteration algorithm or as an approximating Newton method. A detailed derivation and description of the algorithm can be found in [24].

The fundamental iteration step in FastICA for a unit length row vector $\mathbf{w}$ of $\mathbf{W}$, the estimate of the inverse of the mixing matrix, is

$$\mathbf{w} \leftarrow \mathrm{E}\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} - \mathrm{E}\left\{g'(\mathbf{w}^T\mathbf{z})\right\}\mathbf{w} \tag{2.35}$$

where the nonlinearity g can be almost any smooth function. The FastICA algorithm requires the data vectors $\mathbf{z}$ to be centered and whitened.

The algorithm can be used to estimate one independent component at a time using a deflationary orthogonalization algorithm. Another approach is to estimate all the components simultaneously using symmetric orthogonalization. The orthogonality constraint for the independent components follows from the

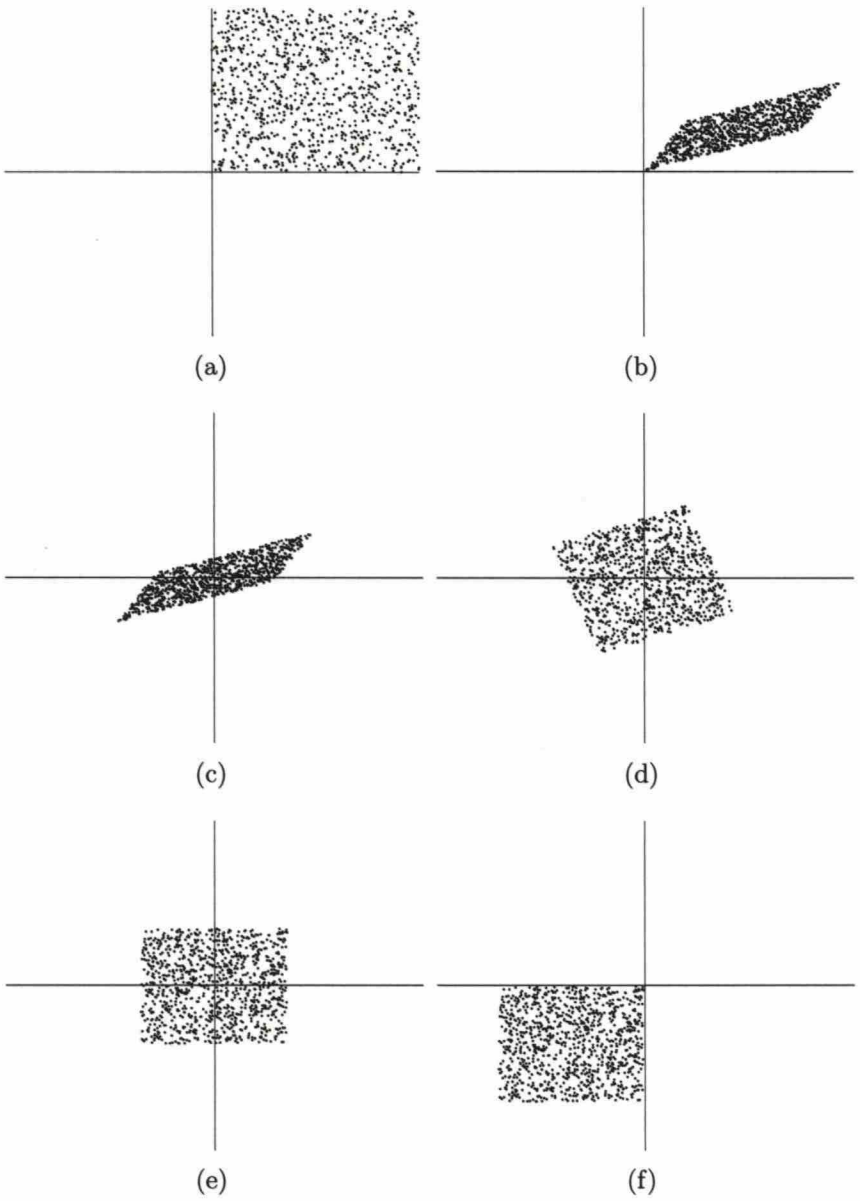


Figure 2.6: Original uniformly distributed two-dimensional data (a) is multiplied by the mixing matrix **A** to create the observed data (b), which is preprocessed by centering (c) and whitening (d). ICA is applied to finding the missing rotation (e), and finally the mean is restored (f). The ambiguities in the signs and the variances of the estimated sources are visible.

independence assumption, which requires that the components are uncorrelated, which equals orthogonality in the whitened space. The constraint is enforced by orthogonalization and normalizing to unit norm the estimated independent components after each step.

Chapter 3

Related works

In this thesis features are extracted from words in contexts using independent component analysis, and the closeness of match between the features and traditional word categories is studied. Earlier, independent component analysis has been applied to several other problems, and different kinds of signals. Word categorization is also a widely researched area and has been approached with several statistical methods.

Statistical methods require input in vector form, which introduces a problem when symbolic written language is studied. Vector space models solve this by representing words as different vectors. In order to model words, contextual information can be calculated from a language corpora. The self-organizing map, latent semantic indexing, singular value decomposition and independent component analysis are statistical methods that have been used to analyze words and text. In the following, these topics will be briefly discussed.

3.1 Contextual information

The use of contextual information is a widely used approach in the statistical analysis of natural language corpora [38, 9, 29]. When a word vector is assigned to each word, contextual information can be calculated as the sum of occurring words in each context. The context can be simply defined as the previous or next word, or a window of words. In information retrieval, each context window could span a whole document. In natural language processing, the frequencies of word sequences of length n, called n-grams, are widely used.

3.1.1 N-grams

A typical task in natural language processing is to calculate the probability $P(V)$ of a word sequence $V = v_1, v_2, \ldots, v_T$. To simplify the calculations, Markov assumption is made: the probability of the k:th word v_k is conditional only to the n preceding words

$$\begin{aligned} P(V) &= P(v_1, v_2, \ldots, v_T) \\ &= P(v_1)P(v_2|v_1)P(v_3|v_1, v_2) \cdots P(v_T|v_{T-1}, v_{T-2}, \ldots, v_2, v_1) \\ &\approx P(v_1)P(v_2|v_1) \prod_{k=3}^{T} P(v_k|v_{k-1}, \ldots, v_{k-n}). \end{aligned} \tag{3.1}$$

The conditional word $P(v_k|v_{k-1}, \ldots, v_{k-n})$ probabilities can be interpreted as Markov-chain transition probabilities. The n-gram model is then the $n-1$:th order Markov model.

The conditional word probabilities need to be estimated from corpus data. The frequencies of word n-grams give maximum likelihood estimates. When n is high, it takes a huge text corpus to get reasonable estimates, because most of the word sequences occur only very infrequently. Bi-grams ($n = 2$) and trigrams ($n = 3$) are commonly used in natural language processing tasks, as they can be estimated with some accuracy from a reasonable amount of data.

3.2 Vector space models

Written language is usually represented using discrete symbols, such as characters and words, in computerized form. Using the symbolic form of, for instance words, directly in numeric algorithms may generate unwanted correlations. For instance, the the semantic closeness between words "window", "glass" and "widow" differ from the correlations between their phonetic or written representations [20].

A general method for representing words in numeric form is to assign an n-dimensional vector $\mathbf{v}_i$ for each word w_i, where the word vectors are orthogonal [41]

$$\mathbf{v}_i^T \mathbf{v}_j = 0, \quad i \neq j. \tag{3.2}$$

To represent a set of words a simple weighted sum

$$\mathbf{c} = \sum_i b_i \mathbf{v}_i \tag{3.3}$$

of the word vectors can be taken. The weights can be used to emphasize wanted words.

In a simple bag-of-words model, the dimension of the word vectors is the same as the size of the vocabulary. Each word w_i is represented by a vector with one in component i and others equal to zero. This ensures that the words vectors for different words are orthogonal.

If the dimension of the words vectors $\mathbf{v}_i$ is equal to the size of the vocabulary, the calculations may be computationally demanding when the number of word grows large enough. Another reason to apply dimension reduction techniques is to allow correlation between related words. Dimension reduction can be interpreted as projecting the bag-of-words vectors to lower-dimensional vectors using a projection matrix $\mathbf{P}$, where the columns of the matrix are the lower-dimensional vectors for the words.

Depending on the method, the projection matrix might try to preserve the orthogonality equally between different word vectors or to try to preserve it only when the words are not related to each other. Random projection is a method that tries to keep all the words orthonormal. A language corpora can be used to measure statistics for the words and a computational method, such as independent component analysis, can be used to calculate the a vector set for the words that relaxes the orthogonality restraint on related words.

3.2.1 Random projection

Random projection (RP) [26] assigns random vectors for word vectors. This equals to using random columns in the projection matrix $\mathbf{P}$. Compared to statistical methods, random projection is fast and computationally simple, but it does not take into account any correlations between the vectors.

3.2.2 Statistical projection methods

Dimension reduction techniques that take into account the connections between words, use statistics of the actual use of words. The calculated statistics

determine the relations between words, and a vector set minimizing the reconstruction error, such as the mean square error, is chosen.

The statistical methods include latent semantic analysis (LSA) that is used with text documents, independent component analysis studied in this thesis, and non-negative matrix factorization (NMF) [13, 8, 49] In the following, latent semantic analysis is studied in more detail.

3.3 Latent semantic analysis

Latent semantic analysis [13, 32] was first introduced as a tool for information retrieval, where singular value decomposition, explained in Section 2.5.1, is used to represent documents using latent topics. The emerging representation enables the comparison of both document and word similarity, as well as keyword querying.

The documents are represented in a term by document matrix $\mathbf{X}$, where the frequencies x_{ij} for terms t_i (rows of the matrix) are given for each document d_j (columns x_j of the matrix). All structural information of the documents is lost in the representation. Usually only a few hundred singular values (latent topics) are extracted from thousands of documents when the term by document matrix is expressed in the reduced space

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{3.4}$$

Each singular value represents one latent topic in the document collection. The left singular vectors, represented by the term by topic matrix $\mathbf{U}$, define the weights of terms in latent topics. Similarly, the right singular vectors in the document by topic matrix $\mathbf{V}$, defines the weights of the latent topics in each document.

The singular value decomposition can be used to compare terms and documents in the low-dimensional space. The matrix

$$\hat{\mathbf{X}}\hat{\mathbf{X}}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\mathbf{S}^2\mathbf{U}^T \tag{3.5}$$

gives all the dot products of the term vectors (rows of $\hat{\mathbf{X}}$) scaled by the singular values and can be used to compare the similarity of terms. Two documents can be compared by the dot product of their document vectors (columns of $\hat{\mathbf{X}}$) scaled by the singular values. All the comparisons are collected into the elements of the matrix

$$\hat{\mathbf{X}}^T\hat{\mathbf{X}} = \mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{S}^2\mathbf{V}^T. \tag{3.6}$$

New documents d_q, or pseudo documents constructed from query terms, can be transformed into a document vector $\mathbf{x}_q$ by calculating the number of occurring terms. The transformation into the latent topic space is

$$\mathbf{v}_q = \mathbf{x}_q^T \mathbf{U} \mathbf{S}^{-1} \tag{3.7}$$

and the components of $\mathbf{v}_q$ give the weights of latent topics in the document d_q. This can be further compared with other documents with Equation 3.6 to retrieve similar documents or examine the distribution of the latent topics.

After its introduction, latent semantic analysis has been applied to modeling language, and has been shown to match human performance [13, 31].

3.4 Self-organizing map

The self-organizing map algorithm creates a non-linear mapping from the original high-dimensional space to a usually two-dimensional sheet-like grid or map of nodes. Unsupervised learning fits the map to the data, so that the learning results in a map that reflects the topology and distribution of the data in the original space.

A self-organizing map can be used in the analysis, visualization, clustering and exploration of the data, and has been applied to various tasks. Earlier papers, where the self-organizing map has been used in analyzing text, include the analysis of artificially generated short sentences [40], Grimm fairy tales [20] and Finnish verbs [29] using contextual information. Short overviews of those papers are given in Section 3.6.

3.5 Independent component analysis

Independent component analysis has been applied to various multimedia signals, including sound [5, 16], images [6, 21], video [44] and text [28, 7]. It has also been applied to other task, such as stock analysis [1] and biomedical analysis [47, 46, 45]. ICA can be used to find an underlying basis of features (latent variables), or to separate mixed signals (cocktail-party problem). Thus it can be applied to several tasks.

The linear noise-free ICA framework has further been extended into several directions. Noisy ICA adds a noise term in the linear framework [22]. Topo-

graphic ICA relaxes the independence assumption between components [23]. Nonlinear ICA allows the generative model to have nonlinearities [25].

3.6 Word categorization

Word categorization and language modeling has been studied using also other statistical methods than ICA. These methods include self-organizing map, clustering and information theory [43, 40, 20, 29].

In [43] an analysis of the Brown text corpus was conducted using contextual information and singular value decomposition. A baseline experiment calculated the left and right contexts for 47,025 words using the the most frequent 250 words as the context words. The resulting 47,025-by-500 matrix was reduced to a 47,025-by-50 matrix using singular value decomposition. The 47,025 reduced word vectors were then clustered into 200 classes using the group-average agglomeration algorithm Buckshot. The words were classified as the most common syntactic category in each class. All occurrences of a word are assigned to one class. The classification was used to measure the performance of part-of-speech tagging. The ambiguity of words was tackled in further experiments. SVD based approach uses only second order statistics, and ICA, that takes advantage of higher order statistics, can be seen as an extension of it.

In [40] the self-organizing map algorithm was applied to averaged contextual information calculated from artificially generated short sentences of nouns, verbs and adverbs. With a thirty word vocabulary, each word was represented by a 7-dimensional random vector $\mathbf{x}_s$ of unit length. The context of a word was calculated as the preceding and the succeeding word pair, and it was averaged over 10,000 generated sentences. The resulting thirty 14-dimensional average word contexts $\mathbf{x}_a$ were normalized to unit length. The input data, $\mathbf{x} = [a\mathbf{x}_s, \mathbf{x}_a]^T$, to the self-organizing map algorithm consisted of the the scaled (with $a = 0.2$) symbol vector $\mathbf{x}_s$ for each word and the averaged word context vector x_a. The semantic word map was created by writing the word at the map position, where the symbol signal $\mathbf{x} = [\mathbf{x}_s, \mathbf{0}]^T$ gave the maximum response.

A similar study using real world data was done in [20], where the relationships between the 150 most frequent words of the Grimm tales were studied. Each focus word was represented by a 90-dimensional random real vector. The code vectors of the words in a contextual window were then concatenated in to a single input vector $x(t)$. In order to equalize the mapping for the selected 150 words statistically, and to speed up computation, a solution used in [40] was

applied. The contexts were averaged relating to a particular focus word. As the main result of the SOM-based analysis, the general organization of the map reflected both syntactic and semantic categories. However, the found categorization is implicit and does not provide explicit features explaining the syntactic and semantic characteristics.

In [29] the conceptual similarities of the 600 most frequent Finnish verbs in a newspaper corpus were studied. Contextual information was calculated using features calculated from the preceding and the two following words relative to the analyzed verbs. The three conducted experiments used the morphosyntactic properties of words, individual nouns in base form and noun categories as features. In all the experiments, random mapping was applied to reduce the dimensionality of the vectors. A self-organizing map with 140 units was taught on the feature vectors for the verbs. The resulting maps were evaluated by comparison with existing verb classification, and by exploring the maps visually. Morphosyntactic features were found to give the best match when compared to existing semantic classification of verbs, and the lexical organization of the map displayed structure based on cultural, social and emotional aspects. Features based on nouns created maps that highlighted topics in the corpus.

Chapter 4

Analysis of words in contexts using ICA

4.1 Introduction

The goal of analyzing words in contexts is to learn interesting and useful components from natural language data with independent component analysis. The learning is done at word level from written language. Here a *word*, e.g. "blue" or "12", is a unique string of letters separated by white-spaces or possibly punctuation marks. The needed statistics are estimated from a text corpus, which represents the usage of written English.

In this thesis, it is assumed that the word usage statistics tell something about the structure and the rules of the language. Word frequencies tell how common a word is. The co-occurrence frequencies of words in contexts contain information on shared features between words.

In linguistics, a syntactic word category is usually defined as a set words, where the syntax of the language is not broken when a word is replaced with another word from the same word category. This is called is a replacement test. For instance, in the sentence "Mary gave John two flowers", the word "flowers" can be replaced with any plural noun without violating the syntax of the language. It is assumed that there is no mechanism for checking whether a sentence is syntactically correct, and a statistical replacement test of context similarity is applied instead. The idea is that, if words occur in similar contexts, they should be assigned to the same category. The co-occurrence statistics are collected into *context histograms*, in which the the co-occurrences of the focus

words in a given context are collected. If there are enough examples of the use of the language, the co-occurrence statistics should tell something about the structure of the language.

It might be reasonable to assume that the context histograms are mixtures of word sets that resemble word categories. As an example, consider what kind of words could immediately precede nouns (e.g. "flower" and "girl"). Here each noun creates a new context histogram, and the counts of immediately preceding words are the elements in the context histograms. The counts of words are calculated for the focus words that are being analyzed. The noun and the location of the focus related to the noun define the context. The mentioned nouns are countable, so they could be preceded by numerals ("two flowers", "four girls"). Nouns can also have other attributes ("beautiful flowers", "young girls"). In case of noun contexts, the histograms might have a high frequency for adjectives. For plural noun contexts, in addition to adjectives, there might also be high frequency counts for numerals.

Word categories are connected to context free grammars (CFGs) and probabilistic context free grammars (PCFGs), which are methods for encoding the syntax of a language, i.e., how sentences are created from connecting smaller elements together [38]. Grammars use rules to show how syntactically correct sentences are created from smaller parts, such as phrases and words categories and words.

A closer connection can be seen between the analysis of words in contexts and Markov models (MMs). With Markov models, there are transition probabilities between words, and word sequences are generated by emitting words using the transition probabilities. With hidden Markov models (HMMs), the transitions are between hidden states and each state has emission probabilities for the observed words. With linear ICA, a weighted sum of the feature vectors encode a histogram of words for each context, for instance, the previous word.

With enough contexts, a statistical feature extraction method might be able to find features that resemble syntactic words categories, for instance, adjectives and numerals. If independent component analysis is used to extract the features, it is assumed that the context histograms are linear mixtures of features.

A numerical representation for the words is needed in order to use them for calculations. This can be accomplished by using a *vector space model* [41] and attaching a real-numbered vector $\mathbf{v}_i$ to each word w_i.

4.2 Context histograms

In the experiments explained in this thesis, the data for the ICA algorithm are the word co-occurrence frequencies in different contexts (context histograms). A context, c, is defined with the surrounding words of the focus word w. The co-occurrence frequencies of words in contexts are the un-normalized maximum likelihood (ML) estimates of the conditional word probabilities $P(w|c)$.

Only a fraction of the the possible focus words and contexts were selected. This was done to decrease the dimension of the problem, and to select a representative set of the context histograms. In the following, the selection process is examined in detail.

4.2.1 Focus words

The number of focus words was limited to the most common terms (different word forms). This was due to the computational costs of too high dimensions and the low frequency counts of rare words. The most common terms contain much of the frequency information and cover a major part of the language so it was natural to chose those words to be modeled. The focus words can be chosen more specifically, for instance, one could be interested in studying only words belonging to a certain category.

4.2.2 Context words

The context words should be chosen to capture as much of the interesting information as possible. This thesis concentrates on examining the similarity between the learned features and syntactic categories and the data will be selected to support that task.

The context words can be selected in many ways, e.g. by examining word frequency [35], function words, variance [37] or by statistical analysis of context histogram consistency [36]. In this thesis, the most frequent words were chosen as the context words. It is a simple method but it should give good results [35].

The most frequent words overlap greatly with the so-called function words (determiners, pronouns, auxiliary verbs etc.), that convey much of the syntactic information by binding together other words (verbs, nouns, adjectives etc.) that carry more specific meaning in a sentence.

An intuitive argument in favor of choosing the most common words is that since they are common, their co-occurrence histograms with other words is fairly dense and they represent most of the frequency information. Furthermore, less frequent words as context words might have more of a semantic role, and their co-occurrence histograms are significantly sparser. For an example, consider the portion of nouns, that might occur in the context of the adjective "humid". The linear ICA might not find traditional syntactic categories, but more semantic ones. If the goal is to find components with more semantic information, the context words could be selected differently.

4.2.3 Context types

The context of a word can be defined in many different ways. In information retrieval, the context might span the whole document and the counts of words are calculated as the occurring words inside the document. Similar approach is used in topic analysis [7], where the text is divided into small overlapping segments which are treated as documents. An example of a more focused context is the n-gram model

$$P(w_k|w_{k-n+1}, w_{k-n+2}, \ldots, w_{k-1}) \tag{4.1}$$

where the next word a sequence, w_k, is modeled using only the $n-1$ preceding words. Simple bigrams ($n = 2$) and trigrams ($n = 3$) are commonly used in natural language processing. The context could also be defined by examining the text in sentence level.

The context can consist of more than one word, or the context word can occur in several places. Examples of different context types would be the the word immediately preceding the focus word ($P(w_k|w_{k-1} = c)$), the two words following the focus words ($P(w_k|w_{k+1} = c_1,\ w_{k+2} = c_2)$), or the context word might appear in a window around the focus words ($P(w_k|w_{k-1} = c \vee w_{k+1} = c)$). Naturally different context types can be mixed simultaneously. Here single word contexts with different locations are used, and the context word and the focus words are not necessarily restricted to be consecutive.

The focus word could be placed far away from the context, or the context words could be distributed apart from each other and the focus word. This kind of approach might capture interesting features, and might even be necessary for some languages, for instance, if morphemes are modeled instead of words. However, this approach will not be taken here, because in English, much of the syntactic information is captured in the neighboring words, when a large corpus is available.

The length of the context must be limited also for practical reasons. Longer contexts would mean higher n-grams and more unreliable probability estimates. It would also mean either having a huge number of context histograms as the combinations of all $n-1$ context words, or creating a method of choosing the context histograms.

Limiting the context length and the position of the focus word, in relation to the context words, restricts the context histograms and the estimated components. It is hoped that enough, if not most, of the syntactic information is captured by the statistics of neighboring words. For better results, it might be a good idea to vary the length of the context and the location of the focus word as much as it is computationally possible. This would mean using context histograms with different lengths and focus word positions.

4.2.4 Creating the data matrix

Contextual information was calculated as context histograms for the focus words. Given a particular context or m chosen words and their positions around the focus word position, the corresponding frequencies were extracted from the text corpus.

When the context histograms have been calculated, a matrix $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_C)$ of size $F \times C$ is created. Here F is the number of the focus words, and C is the number of different context histograms. The rows of $\mathbf{H}$ are the signals for the focus words w_i, $i = 1, \ldots, F$ and the columns $\mathbf{h}_k$, $k = 1, \ldots, C$ are the context histograms. The order of the columns is inconsequential to the ICA algorithm. The creation of the context matrix is illustrated in Figure 4.1, where the focus word and the context create a pattern, whose occurrences in the corpus are calculated.

4.3 Preprocessing

The data matrix $\mathbf{H}$ gives the frequencies of the focus words in each context as the columns. The raw frequency counts are not the best input to the ICA algorithm, because of the large variations in frequencies, so some preprocessing is needed.

The frequency data is concentrated on the most frequent words. To lessen the

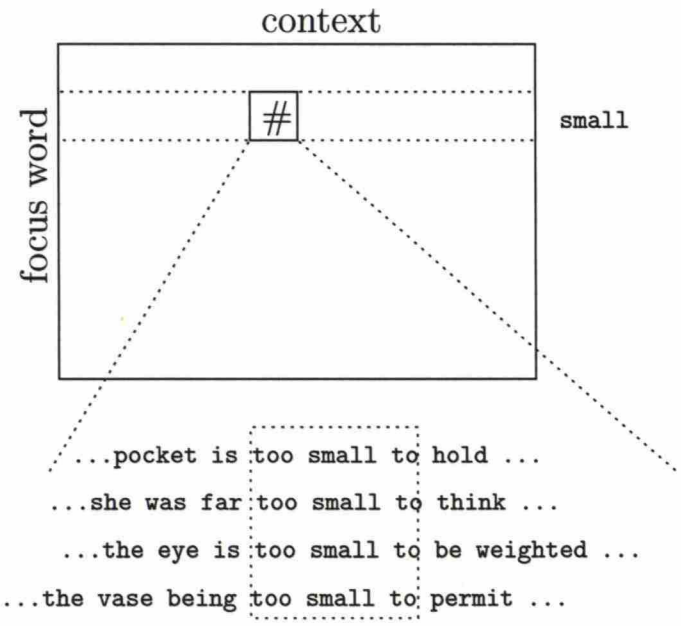


Figure 4.1: The context matrix **H** counts the co-occurrences of the focus words in contexts. The highlighted row is for the focus word "small" with the marked element counting the instances where the focus word follows the word "too" and precedes the word "to" in the corpus. Four of such instances is shown.

effect of the word frequency, the logarithm of the elements increased by one

$$h_{ij} \leftarrow \log(h_{ij} + 1) \tag{4.2}$$

can be taken or the rows of the histogram matrix can be normalized. This kind of preprocessing should make the ICA algorithm model the words more equally, instead of modeling only the most frequent ones.

Normalizing the sum of the columns of the frequency matrix $\mathbf{H}$ to one would correspond to creating a Markov transition matrix from a context to a word. This approach has the downside of reducing the information contained by the data, as the frequency of the context would mean nothing. Two histograms with similar profiles, but with different context frequencies, would be considered equal. For instance, if two contexts had only co-occurrences with a single word, they would be considered equally important after normalization, even if the frequencies would differ significantly.

For computational reasons it might be necessary to perform dimension reduction before applying independent component analysis. The FastICA package [14] uses principal component analysis (PCA, explained in Section 2.4) to reduce the dimension of the problem before the actual ICA algorithm.

4.4 Interpretation of the components

The estimated mixing matrix $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_K)$ and components $\mathbf{s}_k$ are the result of applying ICA to the context histogram matrix $\mathbf{H}$.

The order, signs and variances of the components $\mathbf{s}_k$ cannot be extracted with ICA, so they can be post-processed. The variances of the components $\mathbf{s}_k$ are scaled to one by the FastICA algorithm. As there is no good reason to change the order of the components, they are left as they are.

The columns $\mathbf{a}_k$ of the mixing matrix $\mathbf{A}$ are the features that linearly model the context histograms. Each component can be seen as a sum of word vectors scaled by the intensity of the word in the component. According to our hypothesis, the extracted features $\mathbf{a}_k$ represent syntactic and semantic information in the corpus.

Positive values in the features are easier to interpret than negative values. The skewness of each component $\mathbf{s}_k$ is measured because positive components would have positive skewness. If the skewness is not positive, the component $\mathbf{s}_k$ and the corresponding feature $\mathbf{a}_k$ are multiplied by -1. This does not

affect the ICA model because of the sign ambiguity, but changes the sign of the skewness. The skewness of the component, and not the components, is examined because the components s_{kj} tell the amount of each component $\mathbf{a}_k$. If the skewness of the components were examined, the result would depend on the vector model used for words. This post-processing method assumes that the extracted features are non-negative with possibly signs changed. The assumption is not exactly true, but the post-processed features are very close to being non-negative.

The rows of the mixing matrix $\mathbf{A}$ give a K-dimensional vector representation for each word. If the components $\mathbf{a}_k$ represent linguistic features of the language, such as being a noun or past tense, the words could be analyzed using the rows of the mixing matrix as weights of features for words.

Chapter 5

Datasets

A natural language text corpus provides examples of how language is used. Word usage statistics in contexts are calculated from a collection of electronic texts, called the Gutenberg corpus. Manual syntactic categories attached to words are extracted from a subset of a tagged Brown corpus. In the following, the corpora and preprocessing techniques are introduced.

5.1 Gutenberg corpus

The natural language corpus as the data for the experiments was a collection of English texts from Project Gutenberg[1]. The Gutenberg archive contains copyright-free e-books and e-texts, published mostly before 1923, which can be downloaded free of charge. The language, subject and style of language varies, including plays, novels, letters and text books from various authors.

5.2 Brown corpus

A subset of the tagged Brown corpus[2] was used to extract possible categories for words by listing the tags that were assigned to each instance of the word. In a tagged corpus, each instance of the words is labeled with a syntactic word category. The tag-set consisted of 86 tags, some of which are listed in Table 5.1. The limited vocabulary in the analysis did not take into account tags without

[1] `http://gutenberg.net`

[2] Available on-line at `http://www.ldc.upenn.edu/`

any word instances. The set is a subset of the whole Brown corpus tag set[3], as the used subset of the Brown corpus did not contain all possible tags.

Syntactic word categories for each tag t are represented as vectors $\mathbf{l}_t$ of the same length as the vocabulary. For each word w_i that is marked with the tag t in the tagged Brown corpus, the vector $\mathbf{l}_t$ has one in the i:th element and zero otherwise.

Table 5.1: Examples of the tag-set.

Tag	Description	Examples
.	sentence terminator	. ? ; ! :
AT	article	the a an no every
BEG	verb "to be", present participle or gerund	being
CD	numeral, cardinal	one two hundred 1 ten 1885
DT	determiner/pronoun, singular	this each another that
HVD	verb "to have", past tense	had
IN	preposition	of to in with for at on by
JJ	adjective	great well little good old
JJR	adjective, comparative	better later longer further
MD	modal auxiliary	would will could may should
NN	noun, singular, common	time man day life good king
NNS	noun, plural, common	men people years eyes things
PN	pronoun, nominal	one nothing something none
RB	adverb	some upon now great after
VB	verb, base: noninflected present, imperative or infinitive	see come know make go say take work
VBD	verb, past tense	said came found saw gave
VBN	verb, past participle	said come found seen given

5.3 Preprocessing

The 256 English texts retrieved from Project Gutenberg archives were preprocessed by first removing portions related to the project identification, and concatenating all the texts. Further preprocessing steps consisted of lowercasing all characters, conjoining punctuation marks, and removing most of

[3]All the tags are listed at `http://www.scs.leeds.ac.uk/ccalas/tagsets/brown.html`

the non-alphanumeric characters. The resulting corpus had 21,951,835 word instances in running text and 188,386 unique words forms.

For compatibility, the words in the Brown corpus were also lowercased when listing words for each syntactic category.

5.4 Context histograms

In the experiments, the CMU Language Modeling toolkit [11] was used to extract vocabularies and n-grams from the corpora. Context data was calculated from the n-grams by selecting a single word position inside the n-gram as the focus word, and the rest as the context. This gave a context of length $n - 1$ with n-grams. The focus words and possible context words were limited in the experiments.

Chapter 6

Experiments

The experiments reported in this chapter were performed to experimentally validate the assumption that ICA can find interesting and useful features from natural language text data. Different preprocessing methods and parameters were tried to improve the results. Also, contextual information was calculated in different ways to test its influence on the estimated features. Independent component analysis and singular value decomposition were compared as feature extraction methods to verify that ICA improves the results. The match between syntactic word categories and extracted features was used to compare different features sets. The steps in the experiments are explained in detail in the following, followed by the main experiments.

6.1 Testing procedure

The steps in the experiments can be divided into several steps. First, focus words, context words and contexts types were selected. Next the context histograms were extracted from the Gutenberg text corpus using the selected words and contexts. The frequencies in the context histograms were preprocessed, and ICA or PCA was applied to extract features. The learned features were post-processed and analyzed.

6.1.1 Context histogram extraction

Context histograms were extracted from the Gutenberg corpus of written text. Selected statistics were calculated from the data by choosing focus words and

their contexts. The context histogram extraction is explained in detail in Section 5.4. The context histograms restrict the modeled words and the data for the feature extraction step.

Focus words

The focus words are the words we are modeling and they were chosen to be the most frequent words in the Gutenberg corpus, with the restriction that they must also be present in the Brown corpus. The restriction helped the analysis of the focus words based on the Brown corpus. Selecting the most frequent words makes sure that there are many non-zero entries in the context histograms. The focus words could also be chosen differently, for instance, if only adjectives were chosen, the structure inside adjectives could be examined.

Contexts

Choosing the contexts define the data we are using. The chosen contexts were single left or right word contexts, were there is a single context word either on the left or the right side of the focus word, with no words in between them. Some experiments were conducted with larger contexts with both the left and the right context words. The contexts were calculated using the most frequent words with similar reasoning as in choosing the focus words.

6.1.2 Context histogram preprocessing

The frequencies in the context histograms are non-negative and concentrated on the most frequent words. It is a common practice in natural language processing to reduce the differences in frequencies. An extreme technique is to reduce the frequencies to binary data with one stating that a focus words occurs in the contexts and zero that is does not. Here a less aggressive method of taking the logarithm of the frequencies added by one is used.

6.1.3 Feature extraction

Either independent component analysis or principal component analysis was applied to the preprocessed context histograms $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_C)$ to extract a selected number of features.

6.1.4 Post-processing

In the experiments the components of the learned features $\mathbf{a}_k$ have either high positive or negative components. To take an advantage of this, and the fact that ICA cannot find the signs of the sources, the skewness of the corresponding source $\mathbf{s}_k$ is examined and forced to be non-negative. If the skewness is negative, the source $\mathbf{s}_k$ and the corresponding feature a_k are multiplied by -1. After the post-processing, negative components in the features $\mathbf{a}_k$ have low absolute values compared to positive components. This can be interpreted as emerging "soft" non-negativity, where negative components are low in magnitude compared to positive components. With PCA, the above reasoning doesn't apply. The post-processing was still done, but a greater care was taken in analyzing also the negative correlations.

6.1.5 Feature analysis

The learned features $\mathbf{a}_k$ model the context histograms. With the soft non-negativity of the features in mind, the component a_{ik} can be loosely interpreted as the degree of membership for word i in feature $\mathbf{a}_k$.

To examine the features, the match between features $\mathbf{a}_k$ and manually determined syntactic categories $\mathbf{l}_1, \ldots, \mathbf{l}_T$ is measured, where T is number is syntactic categories. The binary category vectors $\mathbf{l}_t$ are encoded similarly to the context histograms vectors, with one for words w_i in the category, and zero for words outside the category. The syntactic categories are extracted from the tagged Brown corpus, as explained in Section 5.2.

The closeness of match is measured with the normalized correlation

$$c_{kt} = \frac{\mathbf{a}_k^T \mathbf{l}_t}{\|\mathbf{a}_k\| \|\mathbf{l}_t\|}, \quad -1 \leq c_{kt} \leq 1 \tag{6.1}$$

between the feature $\mathbf{a}_k$ and the category $\mathbf{l}_t$. Negative correlations are ignored, and the higher the correlation is, the better the match is. The correlation measure can be interpreted as the cosine of the angle between two vectors in the word space, where the collinearity of the vectors represents similarity. In all the experiments, the correlation with the best matching feature $\max_k c_{kt}$ was selected to represent how well the syntactic category t was found by the feature set. The overall performance of the feature set was measured by the mean of the best matching correlations. Figure 6.1 shows an example feature that has a good match for the VBD and MD categories.

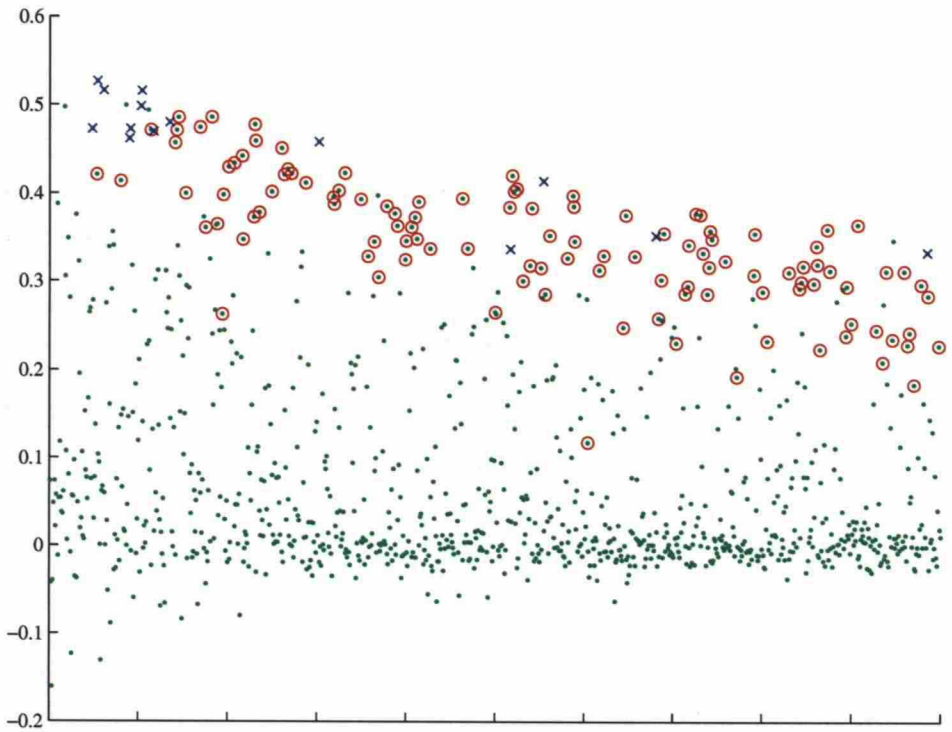


Figure 6.1: An example feature with high correlation with word in VBD (○) and MD (×) categories. The component value (y-axis) for words (x-axis) is marked with (green) dots (·)

An experiment with categories of random collections of words revealed that there is a positive correlation between the number of words in the category l_t and the magnitude of the correlation measure. A more important results was that the ICA-based features do have significantly higher correlations with the syntactic categories than randomly chosen subsets with same number of words, from which can be concluded that the learned features have encoded the syntactic word categories to some extent.

A comparison between the ICA-based features and features extracted with principal component analysis is done in the following experiments and show that the ICA-based features perform better with the correlation measure than the PCA-based features. A comparison to random projected features is not shown, because the dimensions would not encode any linguistic information due to the stochastic nature of the dimension reduction process.

6.2 Experiment with different nonlinearities

This experiment tries to optimize the nonlinearity g in the FastICA algorithm. The non-negativeness of the frequency counts suggests that it might be good to try to find features with positive components. Also, components with only positive signs would be be easier to interpret and compare to traditional syntactic categories.

The function G in the FastICA algorithm is a contrast function, and can be chosen to be almost any non-quadratic function. The derivative of $g = G'$ is called the activation function and appears as a nonlinearity in the fixed-point iteration of Equation 2.35.

In order to combine the robustness of the tanh-nonlinearity and skewness, the *skewed* tanh nonlinearity

$$g(u) = \begin{cases} \tanh(u), & u < 0 \\ a \tanh(u), & u \geq 0 \end{cases} \tag{6.2}$$

was implemented for the FastICA package, where the positive constant a controls the skewness. The larger the value of a, the more skewed the activation function becomes. A value $a = 4$ was found to give a good balance between the skewness and the tanh-nonlinearity.

The new nonlinearity was tested against the nonlinearities 'skew' and 'tanh' provided by the FastICA package. The nonlinearities were tested by extracting a range of features, and calculating the correlation measure for each set of

features. A single left-word context (context type "cw") was used, and the the features were calculated for the most common one thousand words. The variance in the results was taken into account by learning each set of features five times. The mean and one standard deviation of the correlation measure are shown in Figure 6.2. The results show that the skewed tanh performs better than the other nonlinearities, when the correlation measure is used. The correlation measure prefers features with positive components, with negative components reducing the possible highest correlation achievable.

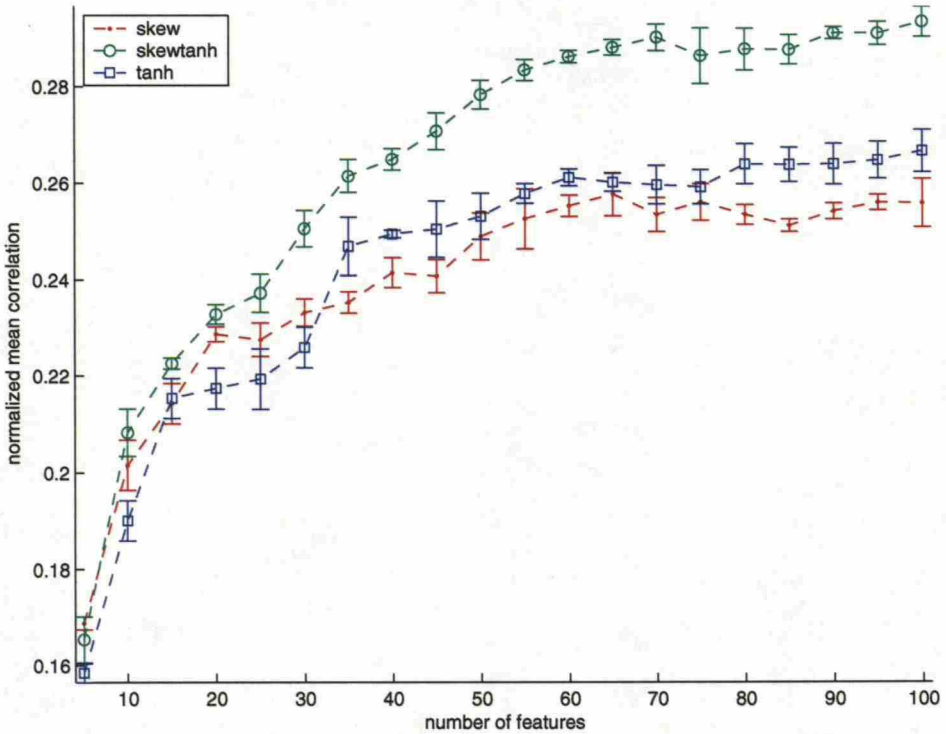


Figure 6.2: Mean and one standard deviation (y-axis) for the best matches between traditional categories and a set of estimated features (x-axis) using the skew (·), tanh (□) and skewed tanh (○) nonlinearities in the FastICA algorithm.

6.3 Preprocessing of frequency counts

In this experiment, the effect of preprocessing the frequency counts was studied. In all the other experiments, the logarithm of the frequencies increased by one was done as a preprocessing step.

The frequency counts of the words in contexts have a wide range of values, depending on the frequency of the focus word and the context words. The dependency is two-fold. First, focus words that are the most common words occur often with many of the context words. Secondly, the context histograms, which have the most common words as the context words, contain most of the frequency information. Both of these effects can be seen in Figure 6.3, which shows the unprocessed context matrix $\mathbf{H}$ for context type *cw*. The focus words (rows) and the context histograms (columns) are approximately ordered by the word frequencies.

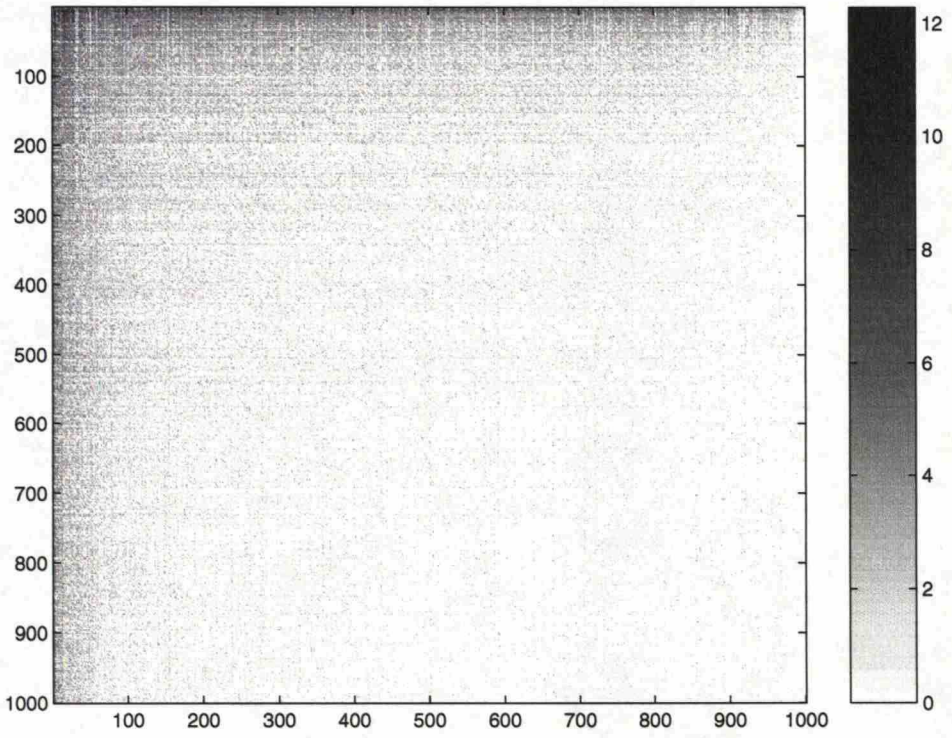


Figure 6.3: The context histogram matrix $\mathbf{H}$ for the most frequent one thousand focus words (rows) and one thousand context words (columns) with context type "cw". The matrix is preprocessed by taking the logarithm of the elements increased by one. The structure and sparsity of the matrix is visible.

It seemed necessary to carry out some kind of weighting or normalization. Taking the logarithm of the frequencies increased by one is a typical balancing technique. It compresses the frequencies, thus making the difference between the high and low counts less dramatic. Normalization of the context histograms and the the focus word were also considered and experimented with, as well as analyzing the data without any normalization.

The preprocessing techniques created two types of results. Taking the logarithm, or normalizing the focus words by setting the norm of the rows of the context matrix to one, resulted in features that model well those syntactic categories that have many words among the focus words. These categories include noun, verb and adjective categories among others. Without preprocessing the context histograms resulted in features that model well categories that contain the most common words. The categories include the punctuation mark category, TO-category containing only the infinitival to, pronoun categories and categories for different inflections of the verb "to be".

If no preprocessing is done, the resulting features seem to concentrate on the most common focus words. Taking the logarithm reduces this effect dramatically but does not remove it altogether. Normalization of the focus words results in very equal modeling of the focus words, but basically ignores the context histograms with small frequency counts.

The conclusions of this experiment was that both taking the logarithm and normalizing the focus words give better results than raw un-preprocessed frequency counts. The learned features are also easier to interpret and analyze further. Both methods could be applied together, for instance, first taking the logarithm and then normalizing the focus words. However, these preprocessing methods will not model the function word categories, such as the different words of the verb "to be", which might be suitable for some tasks.

6.4 Varying the number of context words

In this experiment, the effect of the number of the single left context ("cw" type context) words to the learned features was examined. The context words were ordered according to their frequency and a different number of the most frequent context words were chosen. The context histograms were preprocessed by taking the logarithm of the frequency counts added by one .

The results show that increasing the number of context words affected linguistic features depending on their linguistic categories. Figure 6.4 shows the mean and one standard deviation for normalized correlation measure over five runs with some representative Brown word categories with different number of context words. The correlation measure keeps increasing for some categories, but there are categories where the peak is achieved with a finite number of context words.

The results indicate that there are word categories that are learned best from

a couple of hundred function words, as well as word categories that need large amounts of data to be learned. The other experiments were conducted before this experiment, but the number of context words used in those is not very far off.

Another experiment with using focus word normalization instead of taking the logarithm gave very different results, where the number of context words had only very little effect on how the individual word categories were found. The focus word normalization gives larger weight to context histograms with high frequencies and it is clear that adding context histograms with small values does not change the results.

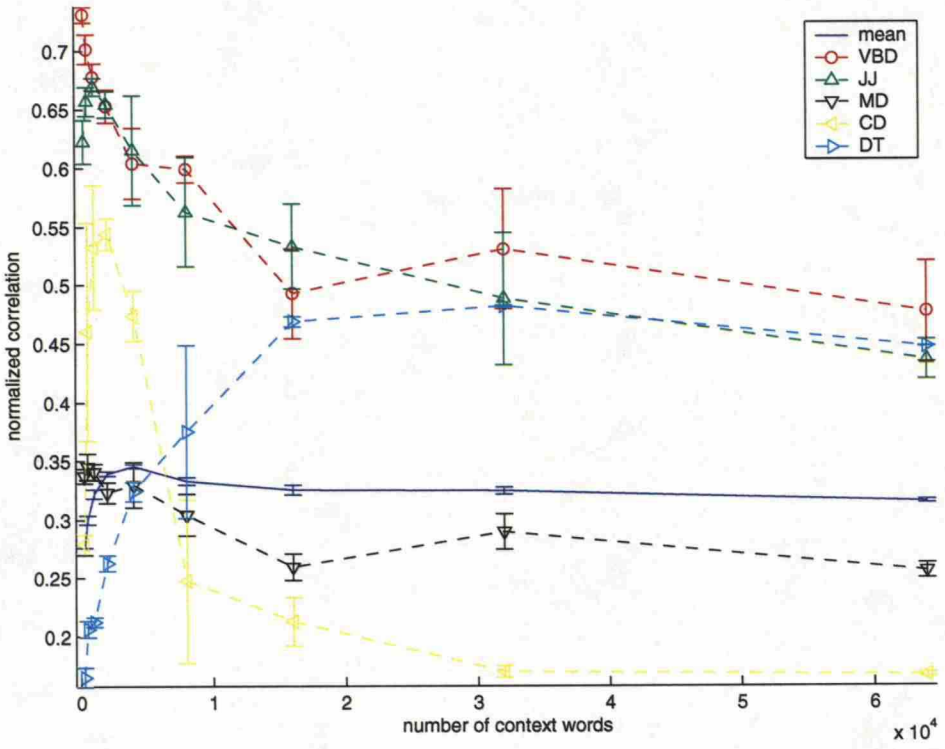


Figure 6.4: Mean and one standard deviation over five runs of the highest correlation measure (x-axis) for representative word categories over ICA features with different number of context words (x-axis). The context type was "cw" and 50 features were extracted for each run. The solid line shows the mean over all syntactic word categories, which has a peak around 3500 samples.

6.5 Varying the context type

One of the main parameters is the context type selection. In this experiment, the main tests for selecting the context type are discussed and the results are shown. The results show that the context type affects the results, as was presumed.

The other parameters used in this experiment were skewed tanh nonlinearity in the FastICA algorithm and preprocessing the data by taking the logarithm of the frequencies increased by one.

The idea behind varying the context types was that longer contexts, i.e., two preceding words instead of only one preceding word, would have more information about the relations between words and would thus give better results. However, longer contexts would also mean sparser data, as there is only a limited amount of data available. Another point to consider is that longer contexts would mean more aggressive pruning of the number of contexts, or else the amount of context histograms could grow quite large. For instance, using two preceding words as the context with N different context words there would be N^2 context histograms, and the memory requirements would grow accordingly. Another approach that was not tested here, would be to concatenate different contexts, for instance single left and right contexts, and the number of contexts would grow linearly.

The experiment was conducted using one and two context words, with different context type positions. For instance, the notation "ccw" states that the two preceding words, the "c"s, were the context words, and "w" was the modeled word. Calculations were conducted with all combinations. However, the word and context were required to be adjacent, i.e., there wasn't any words between the context and the word being modeled.

The number of estimated features was varied, and the mean of the best matches between the syntactic categories and the estimated features was calculated. Figure 6.5 shows the most important results. It can be seen that ICA outperforms PCA with all the context types. As was assumed, longer contexts (two context words versus one context word) resulted in better results. Additionally, from Figure 6.6 it was clear that only the distance between the focus word and context words affected the results. For instance, the "ccw" and "wcc" context types perform equally well, as well as the "cw" and the "wc" context types. The "cwc" context type gave clearly the best results, than the "ccw" and "wcc" context types, which also had two context words but one of the context words was not adjacent to the focus word. The results also show that the overall

performance correlates positively with the number of extracted features. The graphs for features with context types "cw" and "wc" display leveling of the mean correlation as the number of extracted features increases. This suggests that the positive correlation between the mean correlation and the number of extracted features does not depend solely on the number of features.

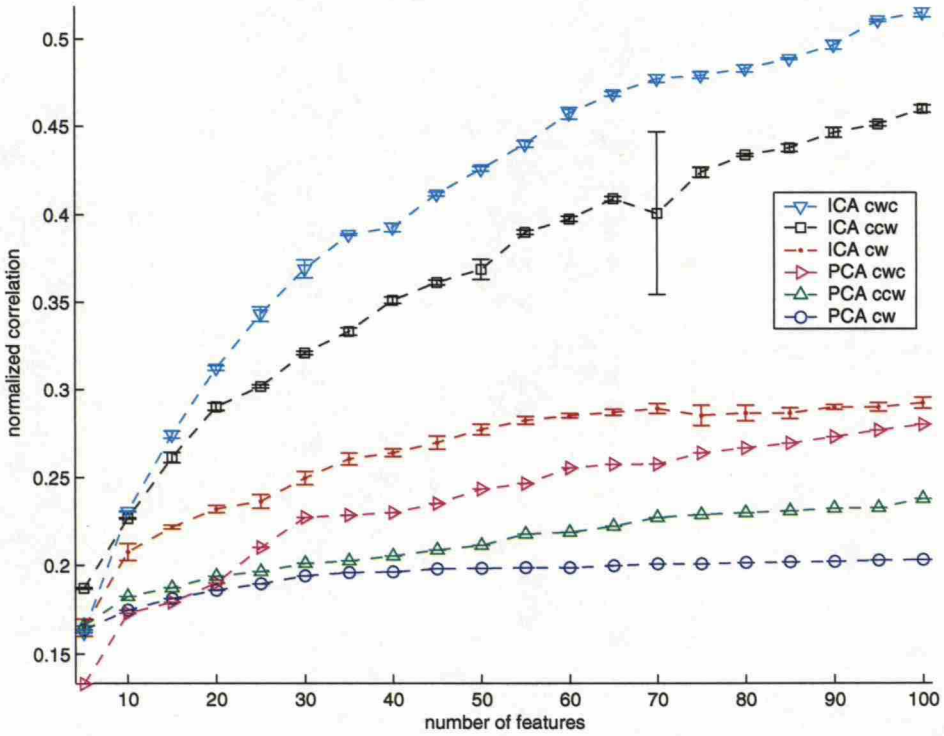


Figure 6.5: Mean and one standard deviation (y-axis) for the best correlations between the traditional categories and a set of estimated features (x-axis). Features were extracted using both ICA and PCA, with different context types. The single relatively large variance at 70 features is caused by a rare failure of the sign-changing scheme.

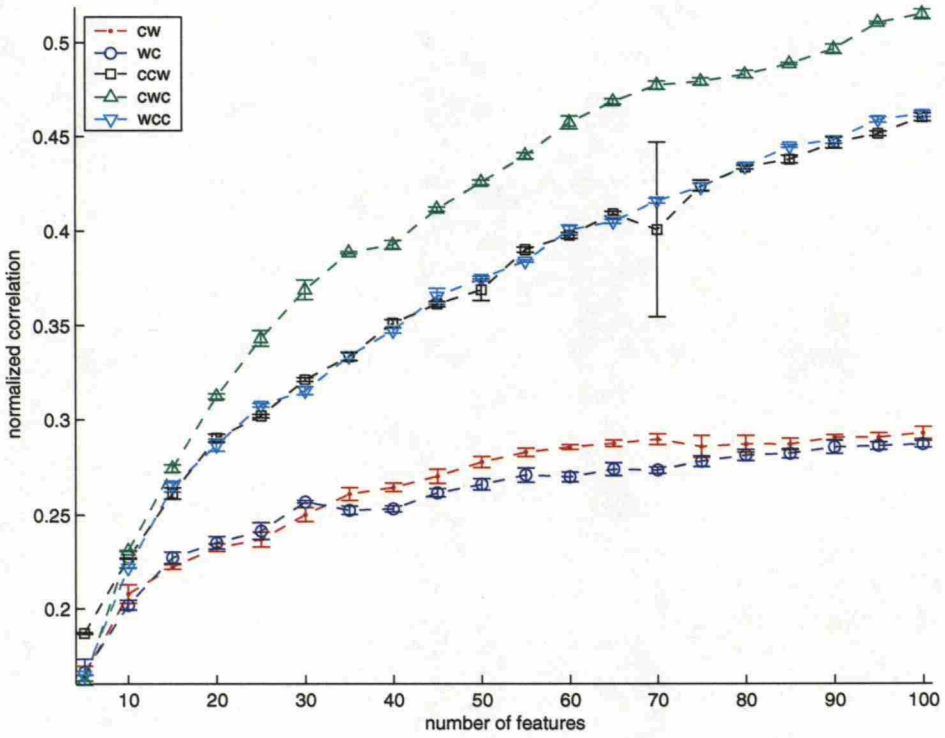


Figure 6.6: The effect of context type to the mean correlation (y-axis) for the best matches with the syntactic categories are calculated for different number of ICA-based features (x-axis). The distances between the context words and the focus word have a major effect on the mean correlation.

Chapter 7

Discussion

In the following, the conclusions of the analysis of words in contexts using independent component analysis are presented. The unsolved questions and possible solutions, as well as aspects of future research are discussed.

In this thesis, the quality of independent component analysis was studied as a feature extraction method for written natural language. Short introductions to independent component analysis and to other methods used were given. The main work related to the analysis of contextual information and using independent component analysis in text analysis was discussed. The contextual information used as data was calculated as occurrences of words in different contexts from a text corpus. The extracted features were compared to traditional linguistic syntactic categories by measuring the correlation between the emergent features and syntactic categories. The meaning of ICA as a word feature extraction method in contexts was discussed. Several experiments were conducted on real world data to study the effect of preprocessing techniques and parameter selections to the results.

The match between the emergent ICA-based features and the syntactic categories was clear in some features, and possible to find in others. The linguistic features emerge as a result of the structure of the language. It was also shown that independent component analysis resulted in a better match between the extracted features and linguistic features, than principal component analysis did. That is, ICA was able to find explicit features representing syntactic categories, whereas PCA resulted in a more implicit representation. The ambiguous nature of natural language causes the syntactic categories to be overlapping, with many word forms belonging to more than one syntactic categories, which made the analysis of the features more difficult.

The most interesting property of the ICA analysis is the ability to find explicit features representing linguistic features based solely on the contextual information, with possibly more than one feature present in a word. Unlike the latent structure found by PCA or SVD, ICA-based features are informative on their own. Also, it is important that words can belong to several categories, i.e., have more than one active feature.

There are unsolved questions left to be studied and answered in the future. It should be experimented whether the linear ICA model can model language sufficiently. For instance, nonlinear ICA might be needed to model the complexity of language. The number of features to extract is an open question in applications of ICA. Part of this constitutes from the uncertainty of not knowing what the extracted features actually model. Modeling the ambiguity of language is another challenge. This thesis treats words in their orthographic form, without handling the possible multiple meanings of each word. This need for disambiguation could be tackled by extracting one or more meanings for each word.

Future work also includes applying the learned ICA-based features to natural language processing tasks, and experimenting with other languages than English. Another future research direction is nonlinear processing, such as thresholding to remove "noise", with the ICA-based features.

Bibliography

[1] A. D. Back and A. S. Weigend. A first application of independent component analysis to extracting structure from stock returns. *Int. J. on Neural Systems*, 8(4):473–484, 1997.

[2] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.

[3] A. Barros, A. Mansour, and N. Ohnishi. Removing artifacts from electrocardiographic signals using independent component analysis. *Neurocomputing*, 22:173–186, 1998.

[4] M. S. Bartlett and T. J. Sejnowski. Viewpoint invariant face recognition using independent component analysis and attractor networks. In *Advances in Neural Information Processing Systems*, volume 9, page 817. The MIT Press, 1997.

[5] A. J. Bell and T. J. Sejnowski. Learning the higher order structure of a natural sound. *Network: Computation in Neural Systems*, 7, 1996.

[6] A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

[7] E. Bingham, J. Kuusisto, and K. Lagus. ICA and SOM in text document analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 361–362. ACM Press, 2002.

[8] J. Charles Lee Isbell and P. Viola. Restructuring sparse high-dimensional data for effective retrieval. In *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.

[9] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[10] A. Clark. Inducing syntactic categories by context distribution clustering. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 91–94, Lisbon, Portugal, 2000.

[11] P. R. Clarkson and R. Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. In *Proceedings of ESCA Eurospeech*, 1997. `http://mi.eng.cam.ac.uk/~prc14/toolkit.html`.

[12] P. Comon. Independent Component Analysis, a new concept ? *Signal Processing, Elsevier*, 36:287–314, 1994.

[13] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[14] The FastICA MATLAB package. Available at `http://www.cis.hut.fi/projects/ica/fastica/`.

[15] S. Finch and N. Chater. Bootstrapping syntactic categories using statistical methods. In W. Daelemans and D. Powers, editors, *Background and Experiments in Machine Learning of Natural Language*, pages 229–235. Tilburg University: Institute for Language Technology and AI, 1992.

[16] F. M. Ham and N. A. Faour. Infrasound signal separation using independent component analysis. In *Proc. 21st Seismic Reseach Symposium: Technologies for Monitoring the Comprehensive Nuclear-Test-Ban Treaty*, Las Vegas, Nevada, 1999.

[17] S. Harnad. *Categorical Perception: The Groundwork of Cognition*, chapter Psychophysical and cognitive aspects of categorical perception: A critical overview. Cambridge University Press.

[18] S. Haykin. *Neural Networks. A Comprehensive Foundation.* Prentice Hall, 1999.

[19] T. Honkela, A. Hyvärinen, and J. Väyrynen. Emergence of linguistic representation by independent component analysis. Technical Report A72, Helsinki University of Technology, 2003.

[20] T. Honkela, V. Pulkki, and T. Kohonen. Contextual relations of words in Grimm tales analyzed by self-organizing map. In F. Fogelman-Soulié and P. Gallinari, editors, *Proceedings of ICANN-95, International Conference*

on Artificial Neural Networks, volume 2, pages 3–7, Nanterre, France, 1995. EC2.

[21] J. Hurri, A. Hyvärinen, J. Karhunen, and E. Oja. Image feature extraction using independent component analysis. In *Proc. NORSIG'96, Espoo, Finland.*, 1996.

[22] A. Hyvärinen. Gaussian moments for noisy independent component analysis. *IEEE Signal Processing Letters*, 6(6):145–147, 1999.

[23] A. Hyvärinen, P. O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13, 2001.

[24] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Sons, 2001.

[25] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

[26] S. Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN'98)*, pages 413–418, Anchorage, Alaska, 1998.

[27] T. Kohonen. *Self-Organizing Maps.* Springer, 1995.

[28] T. Kolenda, L. K. Hansen, and S. Sigurdsson. Independent components in text. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 235–256. Springer-Verlag, 2000.

[29] K. Lagus, A. Airola, and M. Creutz. Data analysis of conceptual similarities of Finnish verbs. In *24th annual meeting of the Cognitive Science Society*, 2002.

[30] G. Lakoff. *Women, Fire, and Dangerous Things.* University of Chicago Press, 1990.

[31] T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquistion, induction and representation of knowledge. *Psychological Review*, 1997.

[32] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[33] M. M. Lankhorst and R. Moddemeijer. Automatic word categorization: An information-theoretic approach. In K. A. S. Immink and P. G. M. Bot, editors, *Forteenth Symp. on Information Theory in the Benelux*, pages 62–69, Veldhoven (NL), May 1993. Werkgemeenschap Informatie- en Communicatietheorie, Enschede (NL).

[34] T.-W. Lee, M. S. Lewicki, and T. J.Sejnowski. ICA mixture models for unsupervised classification and automatic context switching. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 209–214, Aussois, France, 1999.

[35] J. P. Levy and J. A. Bullinaria. Learning lexical properties from word usage patterns: Which context words should be used? In R. French and J. Sougne, editors, *Development and Evolution: Proceedings of the Sixth Neural Computation and Psychology Workshop*, pages 273–282, London, 2001. Springer.

[36] W. Lowe and S. McDonald. The direct route: Mediated priming in semantic space. In M. Gernsbacher and S. Derry, editors, *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*, pages 675–680, New Jersey, 2000. Lawrence Erlbaum Associates.

[37] K. Lund and C. Burgess. Producing high dimensional semantic spaces from lexical cooccurrence. *Behavior Research Methods, Instruments and Computers*, 28(2):203–208, 1996.

[38] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, MA, 1999.

[39] E. Özgen and I. R. L. Davies. Acquisition of categorical color perception: A perceptual learning approach to the linguistic relativity hypothesis. *Journal of Experimental Psychology*, 131(4):477–493, 2002.

[40] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

[41] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[42] E. Sapir. *Language: An Introduction to the Study of Speech.* New York: Harcourt, Brace, 1921; Bartleby.com, 2000.

[43] H. Schütze. *Ambiguity resolution in language learning.* CSLI Publications, 1997.

[44] J. H. van Hateren and D. L. Ruderman. Independent component analysis of natural image sequences yields spatiotemporal filters similar to simple cells in primary visual cortex. *Proc. Royal Society, Ser. B*, 265:2315–2320, 1998.

[45] R. Vigário. Extraction of ocular artifacts from EEG using independent component analysis. *Electroenceph. Clin. Neurophysiol.*, 103(3):395–404, 1997.

[46] R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, and E. Oja. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Advances in Neural Information Processing Systems*, volume 10, pages 229–235. MIT Press, 1998.

[47] J. O. Wisbeck, A. K. Barros, R. G. Ojeda, and N. Ohnishi. Application of ICA in the separation of breathing artifacts in ECG signals. In *Proc. Int. Conf. on Neural Information Processing (ICONIP'98)*, pages 211–214, Kitakyushu, Japan, 1998.

[48] G. Wubbeler, A. Ziehe, B.-M. Mackert, K.-R. Müller, L. Trahms, and G. Curio. Independent component analysis of noninvasively recorded cortical magnetic dc-field in humans. *IEEE Trans. Biomedical Engineering*, 47(5):594–599, 2000.

[49] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM Press, 2003.